

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 13/00 (2006.01)

G06F 15/16 (2006.01)



[12] 发明专利说明书

专利号 ZL 01814750. X

[45] 授权公告日 2006 年 12 月 27 日

[11] 授权公告号 CN 1292358C

[22] 申请日 2001.6.20 [21] 申请号 01814750. X

[30] 优先权

[32] 2000.6.27 [33] US [31] 09/604,426

[86] 国际申请 PCT/US2001/020348 2001.6.20

[87] 国际公布 WO2002/001373 英 2002.1.3

[85] 进入国家阶段日期 2003.2.27

[73] 专利权人 智能网络公司

地址 美国纽约州

[72] 发明人 M·巴克尔

审查员 马晓亚

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 王忠忠

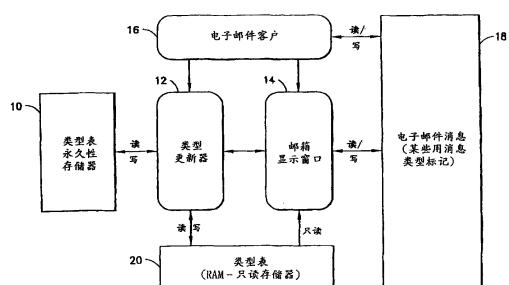
权利要求书 3 页 说明书 44 页 附图 19 页

[54] 发明名称

含有用于识别 MIME 类型及显示不同图标的方法和装置的增强电子邮件系统

[57] 摘要

电子邮件客户软件(16)具有邮箱显示窗口(14)，该邮箱显示窗口列出消息连同每一个消息的图标，在此图标与消息的 MIME 类型相关联。在收件箱中列出含有文件附件(18)的邮件，且带有表明附加到邮件上的文件类型的图标。邮箱显示窗口对 MIME 类型进行解析，并且从 OS 的图标注册表中或从由电子邮件客户软件所维护的图标目录中选择合适的图标。如果在图标目录中没有合适的图标，那么如果 MIME 消息的子部分当中所包含的图标像是可利用的话，则该邮箱显示窗口就使用它。否则，不使用图标或使用普通图标。根据目前优选的实施例，类型表(20)由类型更新器(12)组件来维护。该类型包括一列消息类型，以及连同由邮箱显示窗口使用的可缩放的图标文件名的子类型。



1. 电子邮件客户装置，包括：

a) 显示设备；

b) 邮箱显示窗口，用于在该显示设备上以可滚动列表的方式显示邮箱的内容，其中每个邮件项目被列表且带有多个属性，上述属性从包括主题、发送人姓名和发送日期的组中选出；以及

c) 关联器，用于将多个mime类型与多个图标图像相关联，其特征在于：

上述邮箱显示窗口包括用于确定邮箱中邮件的至少某些项目的mime类型的装置，用于读上述关联器的装置，以及用于在可滚动的列表中显示图标图像的装置，所述图标图像作为与上述邮件的至少某些项目中每一个项目相关联的属性，上述邮件的至少某些项目中每一个项目的上述图标图像是根据上述邮件的至少某些项目中每一个项目的mime类型从上述关联器中选出的。

2. 根据权利要求 1 所述的电子邮件客户装置，进一步包括：

d) 类型更新器装置，用于更新上述关联器从而包含附加的mime类型和附加的图标图像。

3. 根据权利要求 1 所述的电子邮件客户装置，进一步包括：

d) 多个图标图像，每个都具有文件名，其中

上述关联器是将至少某些mime类型中的每一个与图标图像文件名相关联的数据结构。

4. 根据权利要求 1 所述的电子邮件客户装置，其中：

上述关联器包括用于将 mime 类型与程序相关联的装置，并且其中某些 mime 类型不与图标图像相关联而与程序相关联。

5. 根据权利要求 2 所述的电子邮件客户装置，其中：

上述类型更新器装置响应用户输入。

6. 根据权利要求 3 所述的电子邮件客户装置，其中：

上述多个图像中的至少某些图像是可缩放的。

7. 根据权利要求 1 所述的电子邮件客户装置，其中：

上述电子邮件客户装置被设计为与维护图标注册表的操作系统一同使用，

并且

上述关联器将至少某些 mime 类型与从上述图标注册表当中选出的图标相关联。

8. 根据权利要求 1 所述的电子邮件客户装置，进一步包括：

d) 图标恢复装置，用于读取包含于电子邮件或电子邮件附件当中的图形图标信息，其中

如果没有找到其他图标，上述关联器则将 mime 类型与由上述图标恢复装置恢复的图标相关联。

9. 根据权利要求 2 所述的电子邮件客户装置，其中：

上述类型更新器装置可由上述电子邮件客户装置的用户人工地进行操作。

10. 根据权利要求 2 所述的电子邮件客户装置，其中：

上述类型更新器装置根据从包括下列事件的组中所选出的事件来自动安装新的图标图像，该组包括：

安装能够认可/读新 mime 类型的新应用程序或组件，

在接收的邮件消息中遇到新 mime 类型，以及

规律性计划的事件引发上述类型更新器装置为新图标检查服务器。

11. 在显示设备上显示电子邮件箱中内容列表的方法，包括：

a) 用以在显示设备上以可滚动列表的方式显示邮箱的内容的显示步骤，其中每个邮件项目被列出且带有多个属性，上述属性从包括主题、发送人姓名和发送日期的组中选出；

b) 用以确定邮箱当中邮件的至少某些项目的 mime 类型的步骤；

c) 其特征在于，所述显示步骤包括这样的步骤：在可滚动列表中显示图标图像，所述图标图像作为与上述邮件的至少某些项目中每一个项目相关联的属性，上述邮件的至少某些项目中每一个项目的上述图标图像是根据上述邮件的至少某些项目中每一个项目的 mime 类型选出的。

12. 根据权利要求 11 所述的方法，进一步包括以下步骤：

d) 将 mime 类型与程序相关联；

e) 当 mime 类型的邮件项目被打开时，执行与 mime 类型相关联的程序。

13. 根据权利要求 11 所述的方法，其中：

上述图标图像中的至少某些图标图像是可缩放的。

14. 根据权利要求 11 所述的方法，其中：

上述显示步骤包括从图标的中央注册表中选出至少某些图标。

15. 根据权利要求 11 所述的方法，其中：

上述显示步骤包括读取包含于电子邮件或电子邮件附件中的图形图标信息。

16. 根据权利要求 11 所述的方法，进一步包括以下步骤：

d) 当执行上述显示步骤时，为了使用而维护一图形图标的存储库。

17. 根据权利要求 16 所述的方法，进一步包括以下步骤：

e) 在上述图形图标的存储库中自动安装新的图标图像，以响应从包括下列事件的组中所选出的事件，该组包括：

安装能够认可/读新 mime 类型的新应用程序或组件，

在接收的邮件消息中遇到新 mime 类型，以及

规律性计划的事件引发上述类型更新器装置为新图标检查服务器。

含有用于识别MIME类型及显示不同图标
的方法和装置的增强电子邮件系统

本申请是序号为09/209162，申请日为1998年12月10日的申请的部分继续申请，其所公开的全部内容在通过引用结合在此。本申请同时与同共未决的序号为〔BAK-007〕的申请相关，其所公开的全部内容通过引用结合在此。

技术领域

本发明涉及电子邮件程序。本发明尤其涉及具有为邮件项目MIME类型(MIME types)的不同类型显示不同图标的邮箱浏览器显示区域的电子邮件程序。
。

背景技术

近年来，电子邮件（“email”）已经在商务、教育以及个人通信领域中广泛应用。特别是在商务和教育领域中，最为便利的电子邮件的特性之一就是，具备将二进制计算机文件附加到电子邮件消息上的能力。这种特性使电子邮件通信者能够快速地共享文字处理文档、数据库文档、电子表格文档、多媒体文档、或实际上由计算机生成的任何类型的二进制文件。然而，还存在某些与附加二进制文件到电子邮件消息上相关的严重限制和不便利之处。

原始的互联网邮件系统，以RFC (Request for Comments) 821和822定义于1982年，其具有许多重要的限制。尤其是，系统未被设计用来在电子邮件消息中承载大量随机数据。实际上，1982年的SMTP (Simple Mail Transport Protocol) 标准，要求电子邮件由单一消息组成，该单一消息仅包含1000个字符(32K的块)或更少字符的行的ASCII字符。

用于互联网消息的MIME (Multipurpose Internet Mail Extensions) 标准，使通过互联网电子邮件系统发送二进制数据的能力成为可能。原始的MIME标准作为一篇互联网Request For Comments公文 (RFC 1341) 被发表，并且于1992年6月批准通过。（参看最近的MIME标准公文的互联网RFCs 2045、2046和2047。）该MIME标准描述了电子邮件消息应该如何格式化以便于成为MIME所

允许的。MIME定义了一套消息标题字段和一套消息编码标准，其被设计用来克服RFC822消息格式中的限制，以及被设计成可经由任何众多的在互联网上使用的传统邮件传输系统传输。（尤其参看，N.Freed和N.Borenstein, Multipurpose Internet Mail Extensions(MIME)的第一部分：Format of Message Bodies, Network Working Group, Request For Comments(RFC 2045)1996年11月。）MIME消息标题字段在RFC 822中扩展了那些定义，并且描述了电子邮件消息的内容及编码类型。在该MIME标准当中允许的编码方案包括“quoted-printable”，和“base64”。另外，还允许三种非编码数据类型。这些被标为“8bit”（8位）、“7bit”（7位）或“binary”（二进制）。应当注意到，传统的网关仍然无法处理二进制数据，并且几乎所有的MIME允许的消息将二进制数据编码为“7bit”，即MIME的默认编码。

而今，在所有主要的电子邮件客户或“User Agents”（用户代理商），例如 Microsoft Outlook and Outlook Express、Netscape Communicator 和 Qualcomm Eudora上执行MIME。然而，只有少部分MIME类型包括有“text/plain”、“text/html”、“multipart/alternative”和“multipart/mixed”能够由这些程序来处理。或许该MIME标准的最重要特性就是，它允许适当地编码任何二进制数据，以及经由邮件网关及交换机的旧式SMTP系统来发送任何二进制数据。诸如那些上文所列出的邮件客户程序被作了修改，从而允许用户将任何文件类型附加到邮件消息上。这已通过以下方法完成(a)包括适当的编码模块用来将任意文件的二进制数据翻译成如“7bit”或“base64”的可接受MIME编码，(b)扩展邮件客户的能力以处理带有设为“multipart”的MIME类型的消息，以及(c)包括由用户指定的文件以作为“multipart”消息的一部分。很多年，邮件客户程序仅仅给用户提供两个选择；他们能够发送简单的文本消息（以“content-type=text/plain”发送）或者他们能够将任何文件附加到简单的文本消息上（以“content-type=multipart/mixed”发送）。

近些年来，上文所列出的程序已经被扩展到允许程序设计者，通过在邮件客户文本编辑器中包括这些特性以及通过发送被设为“text/html”的MIME类型消息，来使用诸如可选择字体和样式等文本格式的基本类型。现如今，Microsoft's Outlook甚至在将Word文件人工的插入到预发送的邮件消息主体之前，通过将Word文件格式转换成HTML语言，从而允许用户使用全特性文本

编辑器Word撰写电子邮件消息。然而，邮件客户程序仍然完全依赖于带有为任何其他文件格式类型而被设为“multipart”的消息MIME类型的文件附件。

如果带有附加的二进制文件的电子邮件的发送人和接收人正在使用相同类型或版本的电子邮件程序，且两种程序均以相同方式完全配置，那么该接收人的电子邮件程序应会自动将适当的解码过程应用到附加的二进制文件，并且应会产生与由发送人附加到电子邮件上的文件相等同的文件。然而，如果发送人和接收人正在使用不同的电子邮件程序，那么接收方可以接收必须由接收方使用独立的解码程序进行解码的文件。

甚至在文件被完全接收和解码之后，对于收件人而言打开文件通常也是困难的。收件人可能希望通过在文件图标上的“clicking”（点击）就可以打开该文件。然而，在文件图标上的点击经常无法打开文件。这会导致类似于“application not found”（应用程序未找到）等错误消息，更糟糕的情况下，会导致此文件被不适当的应用程序打开从而显示“gibberish”（乱码）。收件人必须具备能够读取（打开）此文件的程序。例如，如果某个人将电子表格附加到电子邮件消息上，则收件人必须具备电子表格程序以便于打开该文件。从技术上讲，收件人并非必须具备与创建该文件的程序一样的类型程序。然而，使用那些不是用来创建该文件的程序来打开该文件，虽然是可能，但会非常困难。收件人必须要知道附加到电子邮件消息上的文件是什么类型的文件，必须要知道他们计算机上的什么程序能够读取那种类型的文件，必须要运行该程序，必须要在该程序当中打开此文件，以及当程序翻译此文件时进行等待。

如果发送人和接收方没有使用相同的操作系统（OS），则互联网电子邮件的限制甚至可能变得更有阻碍。某些邮件附件的编码方案（以及文件压缩方案）是依赖操作系统的（OS-dependent），并且电子邮件接收方接收到不可能被解码（或解压缩）的文件是可能的。

电子邮件当中的这些限制使许多人特别是不具备专业技能的计算机用户，在将文件附加到电子邮件消息上时而气馁。实际上，对于某些初学者用户来说，运行某一应用程序来创建文档、保存文档的任务，和运行独立的电子邮件应用程序以创建电子邮件消息的任务，以及接着将用作附件的已保存的文档定位到电子邮件消息的任务，都十分令人畏惧而令他们气馁。另外，初学者用户时常抱怨，在“downloading”（下载）附加到电子邮件消息上的文件之后，他们在

其硬盘上找不到该文件。

大部分电子邮件客户软件允许用户在收件箱中按发送人、主题或日期将邮件项目排序，以便于更加容易地定位特定的邮件项目。此外，大部分电子邮件客户软件表明该特定消息是否包含有附加文件。例如，其可由诸如回形针图标、普通文档图标或软盘图标来表示。然而，不管附件的性质而使用相同的图标，且直到消息被打开时才能知道该附件的性质。现有技术图1示出一典型的电子邮件收件箱的实例，该收件箱中的某些邮件项目具有由位于主题名称左侧的回形针图标表明的附加文件。尽管没有在图1当中特别示出，但是本领域的技术人员会意识到，诸如!、a、Â、4等的普通图标也可以被显示于消息主题旁边以表明该消息的不同“properties”（属性），诸如是否其为高优先级的消息、是否你已回复了该消息等等。这些普通图标通常是从“dingbats”字体或其他类似字体中取出的单色字体字符。

在最近版本的主要电子邮件客户程序当中，代表所附加文件的文件类型的图标，在该消息被用户打开之后，被显示于邮件消息的主体部分。这很可能是由于诸如Microsoft Windows或Macintosh OS的计算机操作系统维护将信息与系统公知的每个文件类型相关联的数据。此信息包括图形图标，以及可用于在文件上“open”（打开）、“edit”（编辑）或执行其它少量动作的程序的位置。例如，在Microsoft Windows中系统注册表包括系统公知的每个文件类型的表项目，并且上文所描述的至少某些信息与该文件类型相关联。当用户以“content-type=multipart/mixed”打开电子邮件消息时，为Microsoft Windows建立的邮件客户程序（例如，Microsoft Outlook）就确定该消息的第二部分为附加文件，识别该消息内文本的位置例如Attachment Converted：“c:\attach\aFile.doc”，为与该文件类型“.doc”相关联的图标查看系统注册表，并且在该消息主体内显示图形图标。

在目前的系统中，没有使用MIME类型来将图标关联到文件，而宁可使用文件类型的扩展名。这样，对于将不同软件版本或由不同软件版本创建的文档关联到不同图标上的能力，产生了重大限制。例如，由MS Word创建的所有文档，不管使用的是哪一个Word版本，都具有相同的文件类型（文件扩展名），且结果都与相同的图标相关联。即便是在许多更新版本的文件不能被旧版本的软件来读取的情况下也是成立的。

本申请人之前所结合的母案公开了电子邮件软件，其包括一主电子邮件组件以及许多可安装组件。该可安装组件包括用来创建 / 读取不同类型文档的认可/读组件，以及用来列出不同类型消息或用来在不同风格中列出消息的邮箱组件。该主电子邮件组件为直接与存储器相关联的函数以及电子邮件消息的传递提供基础的图形用户接口，并且还处理所有的捆绑或非捆绑数据，其中需要上述数据以将由认可/读组件创建的消息转换成MIME允许的消息。该认可/读组件像嵌入于电子邮件程序内的应用程序那样工作，并且允许从电子邮件程序内创建诸如电子表格、图形、数据库等特定类型的文档并直接以电子邮件发送。该认可/读组件也允许在没有传统上与附加二进制文件到电子邮件信函上相关的难点的情况下，读取所接收的文档。本发明的写组件把数据传递给将数据打包成MIME允许的消息的主电子邮件组件。当消息被接收时，主电子邮件组件将MIME消息连接在一起（如需要的话）且进行解码，并且将数据发送到与MIME类型相关联的认可/读组件当中。

本申请人之前所结合的母案概括地公开并要求保护了邮箱处理软件，由此在模块化组件电子邮件软件的上下文范围内，在邮箱列表中以不同方式显示不同类型的消息。

有理由相信在本申请人的母案中所公开的某些特性，可应用于任何电子邮件客户软件，并且可以用于提高将文件附加到电子邮件上以及使用附加于电子邮件的文件的处理能力。

发明内容

因此本发明的目的就是给电子邮件程序提供不同类型的图标，其中该电子邮件程序包括收件箱列表，借以显示不同类型的消息和附加文档。

将于下文详细讨论的合乎此目的的根据本发明的电子邮件客户软件，具有一邮箱显示窗口，它列出消息连同用于每一个消息的图标，在此图标关联于此消息的MIME类型。含有文件附件的邮件被列于该收件箱中，且带有表明附加到电子邮件上的文件类型的图标。邮箱显示窗口对MIME类型进行解析，并且或者从操作系统的图标注册表中、或者从由电子邮件客户软件维护的图标目录中选择适当的图标。例如，如果接收到带有ADOBE ACROBAT文件附件的电子邮件，则ADOBE ACROBAT图标将会出现在邮箱列表中的邮件项目列表旁边。另外，如果用如在本申请人的母案中描述的特定的认可/读组件创建消息，则

与该认可/读组件相关联的图标将会显示于邮箱列表当中作为显示邮件项目的一部分。

根据本发明的电子邮件客户装置，包括：

a) 显示设备；

b) 邮箱显示窗口，用于在该显示设备上以可滚动列表的方式显示邮箱的内容，其中每个邮件项目被列出且带有多个属性，上述属性从包括主题、发送人姓名和发送日期的组中选出；以及

c) 关联器，用于将多个mime类型与多个图标图像相关联，其特征在于：

上述邮箱显示窗口包括用于确定邮箱中邮件的至少某些项目的mime类型的装置，用于读上述关联器的装置，以及用于在可滚动的列表中显示图标图像的装置，所述图标图像作为与上述邮件的至少某些项目中每一个项目相关联的属性，上述邮件的至少某些项目中每一个项目的上述图标图像是根据上述邮件的至少某些项目中每一个项目的mime类型从上述关联器中选出的。

根据本发明的在显示设备上显示电子邮件箱中内容列表的方法，包括：

a) 用以在显示设备上以可滚动列表的方式显示邮箱的内容的显示步骤，其中每个邮件项目被列出且带有多个属性，上述属性从包括主题、发送人姓名和发送日期的组中选出；

b) 用以确定邮箱当中邮件的至少某些项目的mime类型的步骤；

c) 其特征在于，所述显示步骤包括这样的步骤：在可滚动列表中显示图标图像，所述图标图像作为与上述邮件的至少某些项目中每一个项目相关联的属性，上述邮件的至少某些项目中每一个项目的上述图标图像是根据上述邮件的至少某些项目中每一个项目的mime类型选出的。

本发明的电子邮件软件已通过参考在本申请人的母案中的电子邮件软件的例子作出了描述，其中该电子邮件软件包括一主电子邮件组件，以及许多个经由应用程序设计接口（API）与电子邮件组件进行双向通信的可安装组件。该可安装组件包括认可/读组件以及邮箱显示窗口组件。根据目前优选实施例，还包括一用于维护图标数据库的组件。

当邮箱被打开、邮件的列表被滚动等时候，该邮箱显示窗口组件的功能被用户调用。邮箱显示窗口组件最好包括现有技术中邮箱显示窗口的所有功能，以及包括为了基于消息的MIME类型显示消息相关信息而查看图标目录的功能

。在Lingo实施例当中，为每个带有以消息的MIME类型和子类型为基础的附加TYPE字段的消息创建数据结构。内部TYPE字段用于将MIME类型关联到图标。另一个实施例，直接使用消息的“content-type”（MIME类型）标题的内容来与图标图像相关联。如果在图标目录中没有合适的图标，那么如果包含于MIME消息当中子部分的图标图像数据是可利用的话，则邮箱显示窗口就使用它。否则，不使用图标或使用普通图标。根据目前优选实施例，类型表由类型更新器组件来维护。该类型表包括一列消息类型和子类型，连同将由邮箱显示窗口使用的可缩放的图标的文件名。本发明优选可缩放的图标，以便于该图标能随着用以显示邮箱内容的所选字体的大小来进行调整。

提供了类型更新器组件的几个实施例。根据第一实施例，图标由用户人工地进行安装 / 移除。根据第二实施例，当模块化的认可/读组件被安装 / 移除时，图标被自动地安装 / 移除。根据第三实施例，只要邮箱显示窗口遇到新的消息类型，就会自动添加新图标。新图标或者从操作系统注册表中检索，或者从消息中嵌入的图标图像数据中检索。根据第四实施例，类型更新器为新图标信息自动查询网络服务器，并且如需要的话下载图标图像数据。

对于本领域的技术人员来说，参照详细的说明书述并结合提供的附图，将更加清楚本发明的其他目的和有益效果。

附图说明

图1 是现有技术的电子邮箱显示窗口屏幕；

图2 是根据本发明的电子邮箱显示窗口屏幕；

图2a是根据本发明可选实施例的电子邮箱显示窗口屏幕；

图3 是根据本发明的电子邮件客户的组成结构方块图；

图4 是举例说明根据本发明的邮箱显示窗口的基本操作的简化流程图；

图5 是举例说明根据本发明可选实施例的邮箱显示窗口的基本操作的简化流程图；

图6 是举例说明用于不同邮件项目特性的图标查找的简化流程图；

图6a是举例说明用于MIME类型的图标查找的简化流程图；

图6b是举例说明用于附件filetype（文件类型）的图标查找的简化流程图；

图7是举例说明类型更新器组件的一个实施例的initialize_TypeTable函数的简化流程图；

图8是举例说明类型更新器组件的一个实施例的read_TypeTable_file函数的简化流程图；

图9是举例说明类型更新器组件的一个实施例的write_TypeTable_file函数的简化流程图；

图10是举例说明类型更新器组件的一个实施例的read_icon_files_to_RAM函数的简化流程图；以及

图11是举例说明类型更新器组件的一个实施例的install_Type函数的简化流程图；

附录简要说明

附录A是根据本发明的邮箱显示窗口的MACROMEDIA DIRECTOR程序清单；以及

附录B是根据本发明的类型更新器的MACROMEDIA DIRECTOR程序清单。

具体实施方式

现参照图2，根据本发明的电子邮件客户软件具有邮箱显示窗口，其列出消息连同每一个消息的图标，其中图标与该消息的MIME类型相关联。包含文件附件的邮件，和表示附加到该电子邮件上的文件类型的图标一起在收件箱当中列出。例如，如图2所示，标题为“Minutes of Meeting”的邮件消息被列出且带有一MICROSOFT WORD图标，其表示该电子邮件消息具有附加于其上的WORD文件。标题为“New Radio Ad”的消息被列出且带有一QUICKTIME WAV图标，其表示在该电子邮件上附加了一音频文件。标题为“New Ad Brochure”的消息具有由ACROBAT图标表示的附加ADOBE ACROBAT文件。同样地，消息“Sales Forecast”被显示且带有一EXCEL电子表格图标；“Customer Database”消息被显示且带有一FILEMAKER PRO数据库图标；以及消息“Year End Accounting”被显示且带有一QUICKEN图标。根据本发明，该邮箱显示窗口解释消息的MIME类型和 / 或附件的MIME类型或文件类型，若有的话，从OS（操作系统）的图标注册表中选择适当图标，或者从由电子邮件客户软件所维护的图标目录中进行选择。

通过之前结合的本申请人的母案中关于该电子邮件软件的举例来描述本发明的电子邮件软件，其包括一主电子邮件组件以及许多可安装组件，经由应用

程序设计接口（API）与该电子邮件组件进行双向通信。可安装组件既包括认可/读组件还包括至少一个邮箱显示窗口组件。图2a举例说明了之前结合的本申请人的母案中的电子邮件软件的一个实施例，其中对表示与邮件消息相关联的认可/读组件的图标进行显示。该实施例还可以从现有技术的电子邮件客户那里接收电子邮件（或发送电子邮件到其上）。图2a举例说明了一邮箱列表，其中的两个邮件项目被示为带有两个图标，也就是回形针和ACROBAT图标以及回形针和PHOTOSHOP图标。该双图标的显示表明该邮件被创建且不带有任何特定的认可/读组件，并且具有由某些其它程序创建的附件，在此例中为ACROBAT和PHOTOSHOP。

如上面所提到的，通过之前结合的本申请人的母案中关于该电子邮件软件的举例来描述本发明的电子邮件软件，其包括主电子邮件组件以及许多可安装组件。根据目前优选的施例，还包括用于维护图标数据库的组件。图3举例说明了这些组件之间的连接关系。

如图3所示，被称为TYPETABLE的数据结构10由被称为TYPE_UPDATER的组件12创建并进行维护。在附录B中举例说明了根据本发明的TYPE_UPDATER的工作实例，此附录B在下文被更加详细地描述。该TYPETABLE的主要目的就是将由MBOX_DISPLAYER组件14来读取。该TYPE_UPDATER 12和MBOX_DISPLAYER14利用函数调用及共享数据结构TYPETABLE在相互之间进行通信。例如，该MBOX_DISPLAYER包括对TYPE_UPDATER组件内部的函数“initialize TYPETABLE”(306, 图6a) 的调用。该MBOX_DISPLAYER利用在本申请人的母案以及或利用之前结合的序号为[BAK-007]的申请中所描述的API，与电子邮件客户软件进行通信。如图3所示，MBOX_DISPLAYER组件14和电子邮件客户软件16还具有双向访问用户邮件箱中电子邮件消息存储库的功能。根据目前优选的实施例，如果消息是由本申请人的母案中描述的那种可安装的应用程序组件创建的话，那么该电子邮件消息就以一可选TYPE字段的方式被存储，其中可选TYPE字段包括从MIME类型中以及消息的子类型标题字段中提取的信息。

当邮箱被打开、邮件列表被滚动等时候，该MBOX_DISPLAYER组件的功能被用户调用。本领域的技术人员将意识到，大多数电子邮件客户软件会提供许多不同的邮箱，诸如收件箱、发件箱、读邮件、发邮件等。MBOX_DISPLAYER

组件14最好包括现有技术中邮箱显示窗口的所有功能，并且还包括查看TYPETABLE的功能（以及如下文更详细解释的那样，以浏览MIME消息的主体）从而为邮箱显示区域中消息标题旁边的显示区域查找适当的图标。根目前优选的实施例，图标作为小型图像文件来存储，例如EPS文件或GIF文件，并且通过TYPETABLE数据结构被指向。为了调节邮箱显示区域当中不同大小的字体的使用，还提供了用于缩放图标图形大小的装置。可以使用三种方法。第一，可以利用标准内插算法来缩放该图像。第二，可以存储及检索带有不同分辨率的图标图像的多重拷贝，从而匹配字体磅值大小的限定值。第三，且目前所优选的方法，借以使用该第一、二两种方法的结合，为每个图标至少存储一幅图像，选择与该字体磅值大小最相匹配的图标，并作为需要来接着缩放以更加匹配的字体磅值大小。

表1 举例说明如何将TYPETABLE数据存储于永久性存储器（例如，硬盘）当中。

| mime类型 | 图标文件名 | Msg处理程序文件名 |
|-----------------------|--------------------------|-----------------------|
| text / plain | c:\kidcode\text.gif | c:\kidcode\txt.dxr |
| x-application / rebus | c:\kidcode\rebus.gif | c:\kidcode\ rebus.dxr |
| x-application / grid | c:\kidcode\ grid.gif | c:\kidcode\ grid.dxr |
| x-application / graph | c:\kidcode\ graph.gif | c:\kidcode\ grph.dxr |
| ... | ... | ... |
| multipart/mixed | c:\kidcode\paperclip.gif | |

表1

表1举例说明了至少五种mime类型。前四个是用于认可/读的可安装组件的mime类型。该可安装组件由“.dxr”文件扩展名表示。在表1当中举例说明的multipart/mixed（第五种）mime类型表明与其利用可安装组件，倒不如利用外部应用程序来创建附件。如上文参考图2a所指出的，普通回形针图标用于将附件与利用可安装组件创建的邮件消息相区分。

表2 举例说明当载入RAM中时的类型表（typetable）数据结构。

| mime类型 | ptr | 图标文件名 | msg处理程序文件名 |
|-----------------------|-----|----------------------|-----------------------|
| text / plain | 20 | c:\kidcode\text.gif | c:\kidcode\txt.dxr |
| x-application / rebus | 21 | c:\kidcode\rebus.gif | c:\kidcode\ rebus.dxr |

| | | | |
|-----------------------|-----|--------------------------|----------------------|
| x-application / grid | 22 | c:\kidcode\ grid.gif | c:\kidcode\ grid.dxr |
| x-application / graph | 23 | c:\kidcode\ graph.gif | c:\kidcode\ grph.dxr |
| ... | ... | ... | ... |
| multipart/mixed | 19 | c:\kidcode\paperclip.gif | |

表2

当数据结构TYPETABLE被载入到RAM当中,其被称为SG_TYPETABLE,并且具有如表2所示的结构。这种结构包括指向图标的指针。该指针在示范性实施例中是一LINGO指定成员 (castmember) 指针。

5 如上文指出的,根据该示范性实施例,当电子邮件消息具有附加文件时,会显示一普通附件图标以及显示一该附件所特带的图标。由RAM中创建于程序分支(fly) (例如,如图6b所示) 上的称为SG_ATTACH_TYPETABLE的数据结构来指向这第二个图标。该SG_ATTACH_TYPETABLE的基本结构在表3当中举例说明。

| 文件扩展名 | 图标指针 | 程序文件名 |
|-------|------|--------------------------|
| .doc | 30 | c:\programs\winword.exe |
| .pdf | 31 | c:\programs\acrobat.exe |
| .html | 32 | c:\programs\netscape.exe |
| .htm | 32 | c:\programs\netscape.exe |
| .xml | 32 | c:\programs\netscape.exe |
| ... | ... | ... |

10 表3

如表3所示,文件扩展名与图标指针及程序路径名相关联,其中该程序用于读取附件。参照图6b在下文将更加详细的讨论,该SG_ATTACH_TYPETABLE利用来自于系统注册表内的图标构造于程序分支上。

MBOX_DISPLAYER组件的目前优选的实施例,将在附录A中详细的介绍,15 附录A类似于之前结合的本申请人的母案当中的附录B。示于此附录A中的代码清单,与本申请人的母案中起始于第287行的代码清单不同,该起始于第287行的代码是主邮箱显示函数的开始。该主邮箱显示函数也由图4中流程图解释说明,现参照图4和附录A,该函数起始于附录A当中第287行且在图4的流程图中从START101开始。在显示邮箱内容之前,建立指向该TYPETABLE的指针以及

相关联的系统特征，如附录A中第293—300行和图4中的103、105所示。不同的邮箱字段，例如消息号码、标题、日期、消息读取指示器，在附录A中第302—307行和图4中107被清除。将用于显示图标图形的“Sprite channels”（子图形通路）(MACROMEDIA图形保持器)，在附录A中第309—311行和图4中107被清除。起始于附录A中第313行和在图4中109，下一个消息列表被显示。组成邮箱显示区域的每一行的单元（例如，消息标题、日期、发送人姓名，以及图形单元）被称作为“属性”。在附录A中第320—326行和图4中111、113，这些属性可以从消息中读取。除邮箱、mime类型及状态之外，如图4中111和附录A中第320—323行所描述的那样当读取时，所有属性会自动被显示。在图4中的113和附录A中第324—326行，读取该邮箱、mime类型及状态属性。根据目前优选的实施例，mime类型的图标还用于显示消息状态。如果消息已经被读取，则图标以灰度来显示。如果消息未被读取，则图标以彩色来显示。查找图标以及将其设置成灰度或彩色的步骤，在附录A中第328—355行进行举例说明。在附录A中第332—334行和图4中115，设定子图形通路计数器。在附录A中第340—343行和图4中117，解析该TYPETABLE数据结构。如果该TYPETABLE没有包括用于这条消息的由mime类型表示的图标，那么就在附录A中第344—346行和图4中119选择一默认图标。如果消息的状态表明它（它的附件）已被读取，那么就在附录A中第348—349行和图4中121将图标设定成灰色色度样式。在滚动邮箱内容列表期间，为快速访问而将被选定的图标添加到图标列表当中。此内容在附录A中第351—352行和图4中123描述了。在附录A中第354—374行和图4中125的主邮箱显示区域函数之其余部分，涉及将文本及图标定位在适当的屏幕位置。如图4中127的判断结果所说明的那样，第315—374行所示的代码根据邮箱内消息的数量进行重复。当没有更多的消息将要列出时，如附录A中第376行和图4中129所示，该主邮件显示区域函数结束。

虽然在代码及流程图中目前没有描述，但是如上文所述，如果在图标目录当中没有合适的图标，那么若可能的话该邮箱显示窗口可以使用包含于MIME消息的子部分当中的图标图像数据。本领域技术人员将会意识到，通过从MIME文件中的位置读取图标图像数据很容易实现这种功能，其中该MIME文件由上文中参照的标准被定义。可以将执行此功能的代码插入到附录A中第325行或附录A中第345行。

如上文所述，附录A的剩余部分实质上与本申请人的母案中附录B的邮箱组件相同，并且其说明书已经在之前所结合的本申请人的母案当中作了充分阐述。本领域的技术人员将会意识到，附录中的代码清单为MACROMEDIA DIRECTOR开发程序所特有，并且还会意识到可以利用不同的开发环境来实现相同的功能。图5举例说明了在更为一般的方式下邮箱显示窗口函数的功能，其能够应用于不同程序设计语言中。

现在参照图5，该邮箱显示区域起始于201以读取消息列表。在202进行检查以确定TYPETABLE是否已被初始化。如果尚未初始化，则在204初始化该TypeTable，其中204将参照图7在下文中更详细地描述。在203邮箱显示区域获取下一个消息列表。对于在203获得的消息列表，在205邮箱继续以获取此列表的属性，并且在207获取属性值。如果在209确定该属性使用图标来显示，则在211调用“get icon”程序（此程序已在图6中举例说明）。涉及本申请的消息属性是(a)消息的MIME类型和(b)是否消息带有文件附件。在图5的流程图中（209），该系统检查被作为诸如#type、#date、#mailbox、#status等LINGO符号来执行的当前消息属性是否由图标来代表。尽管在当前执行方式中，每个属性被分开编码，但是在图5的流程图当中描述的更为一般的实施方式，能够通过检查目标符号诸如#type是否为一列表数据结构中的某个单元，例如在图5中步骤209的Properties_with_Icons=(#type,#has_attachment)而实现。在213该图标被显示。在209如果确定属性不使用图标，那么在215显示该属性值。在217确定是否有此消息列表的附加属性。如果有，则程序返回到205。如果没有，则在219确定是否有更多消息列表将要列出。如果有，则程序返回到203。如果没有，则程序在221结束。

图6举例说明包含在附录A第328—350行中的一般化功能。在图5中211是一般的“get Icon（获取图标）”程序调用。程序起始于301，通过调用程序被提供属性名和属性值。如果在303确定该属性为“mimetype（mime类型）”或遵照LINGO执行方式“#type”，那么在305获得该mime类型的图标，并且在307将该图标的指针返回给调用程序。为mime类型获取图标的实际步骤在图6a中描述。如果在309确定该属性为“has attachment”（带附件），则在311就会为“has attachment”获得图标，并且在307将该图标的指针返回给调用程序。用于Microsoft Windows平台的获取文件附件图标的程序流程已在图6b中描述。作为可选择的

，该TYPETABLE数据结构可以包括用于从文件扩展名映射到MIMEtype及MIMEtype图标的filetype字段。如果以这种方法修改TYPEDATA，那么消息类型和文件类型的图标查找将都要使用该TYPETABLE。当遇到经由先前未知的消息类型或经由先前未知的文件类型时，需要在TYPE_UPDATER中进行相关修改以用于安装filetype/MIMEtype组合。由于文件类型扩展名不像Mimetype那么丰富，因此相同文件类型扩展名可能映射到许多不同MIME类型。例如，如果软件制造商为他们软件的不同版本分配不同的MIME子类型，那么上述情况可能会由于相同软件的不同版本问题而产生。

如果在313确定属性为“message read”（读消息），则在315获得“message read”的图标，并且在307将该图标的指针返回给调用程序。如果在317确定属性为“priority（优先级）”，则在319获得“priority”（优先级）的图标，并且在307将该图标的指针返回给调用程序。通过前述内容，本领域技术人员将会意识到，可以为不同属性显示多种不同图标。

图6a举例说明了包含在附录A第328—346行中的一般化执行方式。在图6中305是“get icon for mimetype”（为mime类型获取图标）函数调用。此程序起始于302，通过调用程序提供minetype。在304确定该TYPETABLE是否已被初始化。如果尚未初始化，则在306实现初始化。初始化程序可以驻留在TYPE_UPDATER组件当中，或者可以调用驻留于TYPE_UPDATER组件中的函数。例如，在LINGO执行方式中，函数Read_TypeTable_File（图8），其为该TYPE_UPDATER组件的一部分，被用于初始化TYPETABLE数据结构。在初始化之后，或者如果在306已确定该TYPETABLE已经被初始化，则在308使用mime类型从该TYPETABLE检索图标指针。参看附录A，第328—342行。在310确定图标指针是否为空（null）。如果是，在312调用“Install message type”（安装消息类型）程序。该“Install message type”程序，其包含于TYPE_UPDATER组件当中，在下文参照图11作了详细解释，该图11举例说明了新的消息处理器以及mimetypes图标的安装。如果该图标指针不为空，则在314确定该图标指针是否指向“Icon not Installed”（图标未安装）。如果属于这种情况，则在316将该图标指针设定为默认指针。在任何一种情况下，都在318将非空（non-null）指针返回给调用程序。

图6b举例说明为电子邮件消息的附件获取图标的示例性流程。起始于320，

读取附件的文件扩展名。(注意在Windows平台上，文件名中“.”后面的三个字母确定了该“filetype”。对于其它平台，诸如Macintosh平台，该filetype和“creator code”(生成器代码)被罗列于文件的“resource fork”(资源派生指令)当中。因而，对于那些平台而言，第一步将是从文件的资源派生指令中读filetype(和creator code))。一旦filetype(或filetype和creator code)已被确定，在322程序就试图在SG_TYPETABLE中找到合适的图标。如果在324确定未找到合适的图标，那么在326程序就试图在SG_ATTACH_TYPETABLE中找到合适的图标。如果在328确定未找到合适的图标，那么在330和332程序就试图在系统注册表中找到合适的图标。(注意对于其他操作系统，图标资源可以被存储在不同的地方，例如，在Macintosh OS中图标可存储在不可见的“desktop”(桌面)文件当中。)如果在334找到了合适的图标，则在336设定指向该图标的指针，并且在338将此指针写入到SG_ATTACH_TYPETABLE中。在340将该图标的指针返回给邮箱显示窗口。参看图6中311和图5中211。如果在342没有找到合适的图标，则不提供图标指针。

现在返回到附录B，第1—26行提供了Type Updater(类型更新器)的全面概述和介绍。该Type Updater包括十一个函数。其中的三个是由该邮箱显示窗口调用的公共函数。这些函数包括：Initialize_TypeTable，Install_Type，以及Uninstall_Type。其余的八个函数是Type Updater中使用的专用函数。这些函数包括：Write_Typetable-file，Read_Typetable_File，Read_Icon_Files_To_RAM，read_iconFile，delete_mimetype，insert_mimetype，delete_filetype，insert_filetype。此Initialize_TypeTable函数已在图7中和附录B中第29—60行举例说明。在图7中函数起始于360，在362(附录B中第39行)将SG_TYPETABLE设定为零。在364将SG_ATTACH_TYPETABLE设定为零(附录B中第43行)。在366(附录B中第44行)将存储于磁盘上的Typetable(类型表)文件读入到该SG_TYPETABLE当中。此read_typetable_file内部函数已在图8中和附录B中第207—252行举例说明。在368(附录B中第46行)执行错误检验，以确定是否任何数据已被载入到RAM当中。如果没有数据被读取，在370(附录B中第47行)将对用户发出警告，并在372(附录B中第48行)返回失败。否则，在374(附录B中第51行)将图标文件读入到RAM当中。在376(附录B中第47行)执行错误检验。如果数据已被成功读取，则在378(附录B中第60行)函数结束。

现在回到图8及附录B中第207—252行，此read_typetable_file函数起始于380（附录B中第218行）。在382（附录B中第222行）将该SG_TYPETABLE设定为零。在384（附录B第224-227行）将typetable.txt文件打开，并在386（附录B中第229行）执行文件打开错误检验。如果检测到打开文件过程中的错误，则在388（和附录B中第230行）产生错误警告，在390（附录B中第231行）将文件关闭，并且在392（附录B中第232行）返回SG_TYPETABLE。如果打开文件未带错误，那么在391读取第一行，并在此开始一循环（步骤394—408），当到达文件末尾处时此循环最终在394结束，在此之后在390（附录B中第231行）关闭文件并在392（附录B中第232行）返回SG_TYPETABLE。如附录B中第235—239行所示，由于LINGO不能逐行读取，因而该LINGO执行方式将整个文件读入一字符串并模拟逐行读取。当需对行进行读取时，该行的第一个单词是mimetype（附录B中第243行）。在396（附录B中第244行）将该mimetype插入到SG_TYPETABLE当中。在398如果被发现没有定义该mimetype，则在400警告用户并且在391读取下一行。如果该mimetype已被定义，则读取该行中的下一个单词（filetype），并且在402（附录B中第245行）将其插入到SG_TYPETABLE当中。读取该行中的下一个单词(iconFileName)，并且在404（附录B中第246行）将其插入到SG_TYPETABLE当中。读取该行的下一个单词（msgHandler），并且在406（附录B中第247行）将其插入到SG_TYPETABLE当中。在400报告于408中检测到的任何错误。如上文指出的，该步骤会继续直到该typetable.txt文件被完全读取。

在图9和附录B中第159—206行描述了函数Write_TypeTable_File。这个函数通过Install_Type函数在将新mimetype和图标添加到该SG_TYPETABLE之后，将SG_TYPETABLE的内容写回到typetable.txt中，其中该Install_Type函数在下文参照图11进行描述。该Write_TypeTable_File函数起始于410（附录B中第163行），并且在412（附录B第168行）第一次检验该SG_TYPETABLE结构是否为空。如果是空的，则在414（附录B中第169行）返回错误消息，且在416（附录B中第170行）返回失败消息。如果该SG_TYPETABLE结构不是空的，则在418（附录B中第173行）创建该typetable.txt文件的备份拷贝，并在420（附录B中第176—180行）创建新的空文件。在422（附录B中第181行）执行错误检验，并且如果在创建新文件的过程中检测到错误，则在424（附录B中第182行）返回

错误消息。在426（附录B中第183行）恢复该备份文件，并在428（附录B中第184行）将返回码设定为“fail”（失败）。在430（附录B中第203行）关闭typetable.txt文件。在432（附录B中第204行）删除该备份，并且在434（附录B中第205—206行）返回此返回码。如果创建新文件时无错误，则在附录B中第187—188行将其打开用于写访问，并且在第191行设定一起始行计数器。在436（附录B中第192行）读取SG_TYPETABLE中的第一个（下一个）表项目。如果在438（附录B中第193行）确定已经达到SG_TYPETABLE的末尾，则在440（附录B中第201行）将该返回码设定为“success”（成功），在430关闭该文件，在432删除该备份并且在434返回此返回码。

直到到达SG_TYPETABLE的末尾，才在442将数据设定为写，以及才在446（附录B中第194行）将该mimetype写入字符串。在446、448、以及450（附录B中第195—197行）分别将filetype、iconfilename、以及messagehandlerfilename添加到该字符串当中。在452（附录B中第198行）将该字符串写入到新typetable.txt文件当中，并且在454（附录B中第199—200行）写入行分隔符。该函数循环到436并且继续执行，直到将SG_TYPETABLE中所有表项目都读取完并将其写入到新typetable.txt文件当中。

图10举例说明目前在附录B第255—290行中的函数Read_Icon_Files_To_RAM。在图10起始于456（附录B中第255行），该函数首先在458（附录B中第259行）确定是否任何一种mimetype都在结构SG_TYPETABLE当中被定义。如果一个都没有，未定义图标，则在460（附录B中第260行）返回错误消息，在462（附录B中第261行）将返回码设定为“fail”，并且在464（附录B中第261—262行）返回此返回码。如果该结构SG_TYPETABLE不为空，则在466（附录B中第265行）该函数将默认图标载入到RAM中并且设定指向该图标的指针。在附录B中第268行将图标指针数与LINGO castNumbers关联起来，并且在第270行为增加castNumbers设定计数器。在468（附录B中第271行）读取该结构SG_TYPETABLE中的第一个（下一个）表项目。如果在470（附录B中第272行）确定没有更多的表项目要读取，则在472设定返回码为“success”，并且在464（附录B中第288行）返回此返回码。只要还剩余表项目，就在474读取mimetype以及在476（附录B中第273行）为该mimetype读取iconfilename（图标文件名）。如果在478（附录B中第274行）确定没有与这种mimetype相关联

的图标，那么在480（附录B中第275行）向该mimetype分配默认图标指针。否则，在482（附录B中第277行）利用函数read_iconfile（附录B中第295—298行）读取下一个图标位图和指针。如果在484（附录B中第278行）确定该图标指针不为零，则在第279行增加此castNum。否则不增加此castNum并且在480（附录B中第280行）使用该默认图标指针。在任意一种情况下，在486（附录B中第283行），与mimetype相关联的图标指针被插入。在第284行该计数器被增加，并且在468（附录B中第285行）从SG_TYPETABLE当中读取下一个表项目。

图11举例说明了函数install_type，其用于为特定mimetype或filetype安装新图标和 / 或消息处理器。该函数起始于600且需要进行输入，其中输入包括图标文件名、消息处理器程序名，以及包括mimetype或包括filetype（附录B中第71-75行）。此输入可以由用户或如下文所述的自动装置来提供。在602（附录B中第78行）将此typetable文件读入RAM，从而创建SG_TYPETABLE。如果在604（附录B中第80行）确定SG_TYPETABLE为空，则在606（附录B中第81—82行）返回一错误。否则，若SG_TYPETABLE存在，则在608（附录B中第87行）从SG_TYPETABLE检索。在600中指定的mimetype（或filetype）如果该mimetype（或filetype）存在于SG_TYPETABLE中，并且如果已经将其与图标及消息处理器相关联，那么在612（附录B中第90行）提示用户是否此mimetype（或filetype）需要被重新定义。如果用户选择NO（否），则在614（附录B中第94行）返回一错误。如果此mimetype之前未被定义或者如果用户选择重新定义它，则在616（附录B中第102行）检测于600中指定的消息处理器程序的有效性(例如,是否其目前位于硬盘或网络上)。如果其无效，则在614（附录B中第104—105行）返回一错误。如果该消息处理器和此mimetype（filetype）是有效的，则在618（附录B中第108行）将他们在SG_TYPETABLE中互相之间建立相互关联。接下来，在620（附录B第110行）检测于600中指定的图标文件名的有效性。如果其无效，则在622（附录B中第112行）指定默认图标，并且在624（附录B中第111行）返回错误警告。在任意一种情况下，如果需要，则在626（附录B中第114行）将指定的图标文件名或默认图标关联于SG_TYPETABLE中的消息处理器和mimetype（filetype）。在628（附录B中第122—125行）将Typetable.txt文件从SG_TYPETABLE中写回，并且除非遇到错误，否则在630（附录B中第126行）返回success。

本领域的技术人员将会意识到，该install_type函数可以被其它程序所调用，以至于当安装 / 移除模块化认可/读（modular authoring/reading）组件时，图标可以自动被安装 / 移除。虽然此处并没有以代码或附图特别示出，但是本领域的技术人将会意识到，如何通过参照此处的附录B以及本申请人的母案的组件安装代码来执行此第二实施例。

根据第三实施例，该邮箱显示窗口只要遇到新的消息类型，就会自动添加新的图标。该新图标可从操作系统注册表检索，或者从嵌入于此消息中的图标图像数据检索。通过参照此处的附录B以及上文所涉及的MIME标准，本领域的技术人员将知道这个实施例的执行方式。

根据第四实施例，类型更新器为新的图标消息自动查询网络服务器，并且当需要时或有计划时下载图标图像数据。本领域的技术人员将会意识到，从文件服务器上下载数据的自动更新器是众所周知的，并且会意识到可以通过参考公知的自动更新器连同这里的附录B一起来执行此实施例。

在这里已经描述和举例说明了增强的电子邮件系统的几个实施例，其中该电子邮件系统包括了用于识别mime类型及显示不同图标的方法和装置。虽然已经描述了本发明的特定实施例，但是这并不意指本发明仅限于此，而是意指与该领域同样宽范围的以及与说明书中内容相类似的发明将会被允许。因此，虽然公开了特定的代码清单，但是可以认识到能够使用其他代码。例如，虽然附录举例说明了本发明利用MACROMEDIA DIRECTOR LINGO代码的情况，但此发明也能够用基于“MOZILLA”引擎，利用C++来实施，或通过其他许多流行的开发工具来实现。同时，虽然参考本申请人的母案来公开本发明，其中所述母案包含具有可安装的认可/读组件的电子邮件客户，但是将会认识到，此处所公开的用于显示在邮箱列表中代表mimetypes的图标之规则可以应用于其他类型的电子邮件客户。此外，虽然参考存储图标图像的方法（即，缩放图像）公开特定配置，但是将会认识到，也能够使用其他配置。进一步而言，虽然本发明已经示出了涉及在计算机屏幕上进行显示，但是将会认识到，此显示也可以在电视上、在个人数字助理上、在蜂窝电话上、在腕式手表上等等。本领域的技术人员因此将会意识到，在不偏离本发明的实质和权利要求所限范围的情况下，仍然能够对所提供的发明作出其他的修改。

附录A-邮箱显示窗口代码-Page 1

```

1 — MBOX_DISPAYER IMPLEMENTATION
2 —— Code for a mailbox handler for the KidCode electronic mail client.
3 —— This Director MIAW displays a mailbox in a window as a list
4 — of messages with one line for each message. Each message has the following fields displayed:
5 — 1. message number
6 — 2. Message sender
7 — 3. Message mimetype & status (single icon used to indicate both properties)
8 — 4. Message subject header
9 — 5. Message date
10
11
12 on startMovie
13     global SG_LastActiveWindow — super global that keeps track of KC last active window
14     global mbxG_username — current user name
15     global mbxG_messages — list of messages
16     global mbxG_nMsgs — number of messages in mailbox
17     global mbxG_boxName — current mailbox name
18     global mbxG_whichLine — current hilite line = msgNumber
19     global mbxG_subtractLine
20     global mbxG_lips
21
22     set mbxG_lips = 0
23
24     — called by API to Main Movie
25     tell the stage to emh_continue(#mailbox)
26 end
27
28
29 on activateWindow
30     global SG_LastActiveWindow
31     global mbxG_myWindow
32     set SG_LastActiveWindow = mbxG_myWindow
33 end activateWindow
34
35
36 — Stop movie handlers
37
38 StopMovie handler in a MIAW is called only when the movie
39 — plays through to the end or jumps to another movie.
40 — It isn't called when the window is closed or the window
41 — is deleted by the forget window command.
42
43 on stopMovie
44     cleanUpMovie()
45 end
46
47 called to close a MIAW or automatically whenever
48 — forgetWindow is called
49
50 on closeWindow
51     cleanUpMovie()

```

附录A-邮箱显示窗口代码-Page 2

```

52 end
53
54
55 — cleanUpMovie can be called by both stopMovie and
56 — closeWindow.
57
58 on cleanUpMovie
59     global mbxG_whichLine
60     global mbxG_subtractLine
61     global mbxG_nMsgs
62     global mbxG_username
63     global mbxG_lips
64
65 — CLEAR FIELDS AND GLOBAL VARIABLES
66
67     put "" into field "MailboxTitle"
68     put "" into field "prepositionTitle"
69     put "" into field "MessageNumber"
70     put "" into field "MailboxTo"
71     put "" into field "MailboxSubject"
72     put "" into field "MailboxDate"
73     put "" into field "Messageread"
74
75     set mbxG_nMsgs = 0
76     set mbxG_lips = 0
77     resetHilite()
78
79     if findEmpty(member 50) > 50 then
80         set the scriptText of member 50 = ""
81     end if
82
83     set the memberNum of sprite 6 = the memberNum of member "lips up"
84     set the loc of sprite 4 to point(800, 4)
85     set the loc of sprite 5 = point(800, 19)
86 end cleanUpMovie
87
88
89 — API Public Handlers ——————
90 —
91 — Ugly hack to work around problem with Director startup
92 — of MIAWs. The problem is that, after calling a handler in the
93 — MIAW, the StartMovie handler for the MIAW does not run until
94 — the calling movie advances to its next frame.
95 — Therefore, the calling sequence in the calling movie
96 — has to be engineered so that the real handlers in the MIAW do not
97 — run until after control has been transferred back to the calling
98 — movie. However, at least one handler in the MIAW must be called
99 — by the calling movie before the StartMovie handler will run.
100
101 — startMeUp is the fake handler that, when called by the
102 — main movie, will return to the main movie,

```

附录A-邮箱显示窗口代码-Page 3

```

103 — cause this movie's startMovie handler to run.
104
105 — The second part of this wormy hack is contained in the MIAW's
106 — startMovie handler... It is a call to a workAround handler in
107 — the calling movie called continueComponent
108 — The calling movie's continueRebus handler calls the real handlers
109 — in the MIAW.
110
111 on emc_startMeUp
112     — put "Macromedia sucks!"
113     return(TRUE)
114 end emc_startMeUp
115
116
117 initWindow is called by email main when a message handler
118 — is opened
119
120 on emc_initWindow userName, windowName
121     global mbxG_myWindow
122     global mbxG_username
123     global mbxG_platformType
124
125 — put "ENTER emc_initWindow mailbox"
126     set mbxG_username = userName
127     set mbxG_myWindow = windowName
128
129     — puppet the hilite (MB 4-17-99) removed this
130     resetHilite()
131     tell the stage to emh_getColorDepth()
132     set colorDepth = the result
133     mapColors(colorDepth)
134     tell the stage to emh_getPlatformType()
135     set mbxG_platformType = the result
136
137 — put "EXIT emc_initWindow mailbox"
138
139     return(TRUE)
140
141 end emc_initWindow
142
143
144 — closeWindow is not called unless Rebus plays as
145 — a MIAW.
146
147 on emc_closeWindow
148     — put "ENTER emc_closeWindow Mailbox"
149     closeWindow()
150     — step frame
151     — put "EXIT emc_closeWindow Mailbox"
152     return(TRUE)
153

```

附录A-邮箱显示窗口代码-Page 4

```

154 end emc_closeWindow
155
156
157
158 on emc_getComponentInfo
159   return( list( "SimpleMail", 1, #mailbox, "text" ) )
160 end emc_getComponentInfo
161
162
163
164 on mbx_getMessage
165
166   —— “open” button and doubleClick of highlighted message
167   —— calls to email main to hand the message
168   —— selected to a message handling movie
169   —— This script was previously the “open” cast member script:
170
171   —— global mbxG_whichLine
172   —— global mbxG_messages
173
174   —— set mailData = getAt(mbxG_messages, mbxG_whichLine)
175
176   —— return(mailData)
177
178
179
180   ——end mbx_getMessage
181
182
183
184 on mbx_getMessageNumber
185   global mbxG_whichLine
186   return(mbxG_whichLine)
187 end mbx_getMessageNumber
188
189
190   —— mbx_trashMessages returns a list of messages that are to
191   —— be trashed in the mailfile. Email main will rewrite the mail file
192   —— When implemented correctly, it will determine which message numbers
193   —— are associated with the currently selected lines in the mailbox
194   —— display, update the display to remove these messages from the
195   —— list, and return the list of deleted message numbers.
196
197 on mbx_trashMessages
198
199   global mbxG_messages — list of messages
200   global mbxG_nMsgs — number of messages in mailbox
201   global mbxG_whichLine — current hilite line = msgNumber
202
203   —— set mailData = getAt(mbxG_messages, mbxG_whichLine)
204

```

附录A-邮箱显示窗口代码-Page 5

```

205      —— tell the stage
206      — return(mailData)
207      —— end tell
208
209      — needs implementation that can handle multiple messages
210      — also need to rewrite trashIt which does not conform to
211      — API rule that only API handlers can be called in other movies
212
213      — set message = mbxG_messages
214      if mbxG_whichLine > 0 AND mbxG_whichLine <= mbxG_nMsgs then
215          tell the stage to emh_alertUserToTrash()
216          set yes = the result
217          if not yes then return [] -- user canceled
218
219          set trashList = list(getAt(mbxG_messages, mbxG_whichLine))
220          deleteAt(mbxG_messages, mbxG_whichLine)
221          set mbxG_nMsgs = mbxG_nMsgs - 1
222          displayMailbox(mbxG_messages)
223          resetHilite()
224      else
225          alert("Please click on the message you wish to delete.")
226          set trashList = []
227      end if
228
229      return(trashlist)
230
231 end mbx_trashMessages
232
233
234 —— accepts a mailbox datastructure that consists of a boxname and
235 —— a list of messages
236
237 on mbx_openMailbox mailbox
238     global mbxG_username
239     global mbxG_messages
240     global mbxG_boxName
241     global mbxG_nMsgs
242
243 — put "ENTER mbx_openMailbox"
244     set mbxG_boxName = getAt(mailbox, 1)
245     put mbxG_userName & "'s " & mbxG_boxName into field "mailboxTitle"
246
247     set mbxG_messages = getAt(mailbox, 2)
248     set mbxG_nMsgs = count(mbxG_messages)
249
250     displayMailbox(mbxG_messages)
251
252 — put "EXIT mbx_openMailbox"
253     return(TRUE)
254
255 end mbx_openMailbox

```

附录A-邮箱显示窗口代码-Page 6

```

256
257
258
259 Utilities -
260
261 Initialize formatting of text fields
262 — Thanks to Frank Leahy for this one
263
264 on SetTextInfo fldName, fldValue, fldAlign, fldFont, fldSize, fldStyle
265     if fldValue <> EMPTY then
266         put fldValue into field fldName
267     end if
268     set the textAlign of field fldName = fldAlign
269     set the textFont of field fldName = fldFont
270     set the textSize of field fldName = fldSize
271     set the textStyle of field fldName = fldStyle
272 end
273
274
275 on formatFields
276
277 — FORMAT THE TEXT FIELDS
278
279     setTextInfo "MessageNumber", "", "left", "arial", 14, "bold"
280     setTextInfo "MailboxTo", "", "left", "arial", 14, "bold"
281     setTextInfo "MailboxSubject", "", "left", "arial", 14, "bold"
282     setTextInfo "MailboxDate", "", "left", "arial", 14, "bold"
283     setTextInfo "MessageRead", "", "left", "arial", 14, "bold"
284
285 end formatFields
286
287 — MAIN MAILBOX DISPLAY FUNCTION
288 — displays a Mailbox style listing of messages
289 — places the appropriate components from each message
290 — into field members with lines aligned for display
291
292 on displayMailBox msgList
293     global SG_TYPETABLE — super global variable shared across different MIAWS
294     global mbxG_red
295     global mbxG_platformType
296     global mbxG_iconList
297
298     — mbxG_iconList for future use in up/down scroll scripts
299     set mbxG_iconList = [:]
300     set count = 0
301
302     — first clear all the fields and the sprites
303     put "" into field "MessageNumber"
304     put "" into field "MailboxTo"
305     put "" into field "MailboxSubject"
306     put "" into field "MailboxDate"

```

附录A-邮箱显示窗口代码-Page 7

```

307      put "" into field "Messageread"
308
309      repeat with i = 40 to 70
310          set the member of sprite i = member "blank"
311      end repeat
312
313      — Fill the text fields with info from all of the messages so that it is available
314      — when the window scrolls
315      repeat with msg in msgList
316
317          — with the exception of mailbox, mimetype and status,
318          — the fields will automatically be displayed when filled
319
320      put the lineCount of member "MailboxTo"+1 & RETURN after field "MessageNumber"
321      put getProp(msg, #from) & RETURN after field "MailboxTo"
322      put getProp(msg, #subject) & RETURN after field "mailboxSubject"
323      put getProp(msg, #date) & RETURN after field "mailboxDate"
324      put getProp(msg, #mailbox) into mailbox
325      put getProp(msg, #mimetype) into mime
326      put getProp(msg, #status) into status
327
328      — Display the mimetype icon
329      — We use the mimetype icon to also indicate message status. A greyscale version of the
330      — icon is displayed if a message has been read. Otherwise a color icon is displayed.
331
332      — Icons will be placed in sprite channel 40 and beyond
333          set i = 40 + count
334          set count = count + 1
335
336      — find the icon for the message mimetype
337      — this code should be rewritten using a data access function. For now we need to
338      — know the format of the TYPETABLE data structure
339
340          set mimeProperties = getProp(SG_TYPETABLE, mime)
341          set iconCastMember = getAt(mimeProperties, 2) — second item is the cast member
342          number
343
344          — if this type is unknown then use the default icon
345          if iconCastMember = 0 then set iconCastMember = the number of member
346          "DefaultIcon"
347
348          — get the greyscale version of the icon if the message has been read.
349          if status = "R" then set iconCastMember = iconCastMember + 1
350
351          — add the icon to the list of icons used for scrolling the window
352          append (mbx_iconList, iconCastMember)
353
354          — place the icon at the proper location for display
355          set the memberNum of sprite i to iconCastMember
356
357          — display the icon in the correct grid cell in the mailbox message list.

```

附录A-邮箱显示窗口代码-Page 8

```

358      — set the locH of all icons to 50
359      set whereGoesIcon = the lineCount of member "MailboxTo"
360      puppetSprite i, TRUE
361      set the visible of sprite i = TRUE
362      set the locH of sprite i to 50
363
364      — precise placement of the icon next to it's message...
365      set positonVar = 105 + linePosToLocV(member "MailboxTo", whereGoesIcon)
366      set the locV of sprite i to positonVar
367      — but don't let icons fall go beyond window if there are many messages
368      if positonVar > 550 or positonVar < 105 then
369          set the visible of sprite i = FALSE
370      else
371          set the visible of sprite i = TRUE
372      end if
373      addProp mbxG_iconList, (the locV of sprite i), mime
374      end repeat
375
376  end displayMailbox
377
378  — FUNCTIONS USED TO RESPOND TO USER INTERACTION WITH MAILBOX
379  — HILITE MESSAGE is called when a user clicks a mouse on a message line
380
381  on hiliteMessage whichLine
382      global mbxG_nMsgs, mbxG_whichLine, mbxG_subtractLine, mbxG_messages
383
384  — KEEP TRACK OF SELECTED LINE
385
386  set mbxG_whichLine = whichLine
387
388  — MAKE SURE LINE IS VALID
389
390  if mbxG_whichLine <= 0 then
391      return(0) — do nothing, errors are caught elsewhere
392  else if mbxG_whichLine > mbxG_nMsgs then
393      — user clicked somewhere else in field
394      set mbxG_whichLine = 0 — reset to 0
395      return(0)
396  end if
397
398  — HIGHLIGHT SELECTED LINE
399
400  set whichHighlight = mbxG_whichLine + mbxG_subtractLine
401
402  — since all field members in display are kept synchronized
403  — any one will do for linePosToLocV
404  — use "MailboxTo", it's small
405
406  set the locV of Sprite 11 to ~
407      (99 + linePosToLocV(member "MailboxTo", whichHighlight))
408

```

附录A-邮箱显示窗口代码-Page 9

```

409 —when the user double clicks on a hilited message, go get it from mailFile
410
411 if the doubleClick then
412     set maildata = getAt(mbxG_messages, mbxG_whichLine)
413     tell the stage
414         cmh_openMessage(maildata)
415     end tell
416     — mbx_getMessage()
417 end if
418
419 end hiliteMessage
420
421
422
423 on resetHilite
424     global mbxG_whichLine, mbxG_subtractLine
425
426     set mbxG_whichLine = 0
427     set mbxG_subtractLine = 0
428
429     — SET HIGHLIGHT OFF STAGE
430     set the loc of sprite 11 to point (11, -20)
431
432 end resetHilite
433
434 — this is a lookup table for color
435 — only really necessary for Mac platform
436 — use on any color that you want to
437 — set the forecolor of field
438
439 on mapColors colorDepth
440     global mbxG_red
441     global mbxG_blue
442     global mbxG_white
443     global mbxG_black
444
445     case colorDepth of
446
447         8:
448             set mbxG_red = 6
449             set mbxG_blue = 4
450             set mbxG_white = 0
451
452         16:
453             set mbxG_red = 31744
454             set mbxG_blue = 31
455             set mbxG_white = 32767
456
457         32:
458             set mbxG_red = 16711680
459             set mbxG_blue = 255
             set mbxG_white = 16777215

```

附录A-邮箱显示窗口代码-Page 10

```

460      end case
461
462      set mbxG_black = the forecolor of line 1 of member the member of sprite 4
463
464 end mapColors
465
466
467 scripts run when the mouse is clicked on a mailbox message line.
468 A script is needed for each field in the message line.
469
470 on mouseUp
471     hiliteMessage(the clickon - 40)
472 end
473
474
475 on mouseDown
476
477     global mbxG_lips
478     .set whichLine = the mouseLine
479     if mbxG_lips then
480         set astr = line whichline of field "MailboxTo"
481         speak(astr)
482     else
483         hiliteMessage(whichLine)
484     end if
485 end
486
487 on mouseDown
488
489     global mbxG_lips
490     set whichLine = the mouseLine
491     if mbxG_lips then
492         set astr = line whichline of field "MailboxSubject"
493         speak(astr)
494     else
495         hiliteMessage(whichLine)
496     end if
497 end
498
499
500 more scripts run when the mouse is clicked on a mailbox message line
501
502 on mouseDown
503     global mbxG_lips
504     set whichLine = the mouseLine
505     if mbxG_lips then
506
507         set astr = line whichline of field "MailboxDate"
508         set aday = word 1 of astr
509         case aday of
510             "Mon,": put "Monday" into word 1 of astr

```

附录A-邮箱显示窗口代码-Page 11

```

511      "Tue,": put "Tuesday" into word 1 of astr
512      "Wed,": put "Wednesday" into word 1 of astr
513      "Thu,": put "Thursday" into word 1 of astr
514      "Fri,": put "Friday" into word 1 of astr
515      "Sat,": put "Saturday" into word 1 of astr
516      "Sun,": put "Sunday" into word 1 of astr
517      otherwise
518      end case
519
520      speak(astr)
521      else
522          hiliteMessage(whichLine)
523      end if
524  end
525
526  on mouseDown
527      global mbxG_lips
528
529      set whichLine = the mouseLine
530      if mbxG_lips then
531          set astr = line whichline of field "MessageNumber"
532          speak(astr)
533      else
534          hiliteMessage(whichLine)
535      end if
536  end
537
538
539 — this script is attached to the message type icon which is displayed in the message line
540
541  on mouseUp
542      set whichLine = the mouseLine
543      hiliteMessage(mouseLine)
544  end
545
546 —— code for scroll buttons
547
548  on mouseDown
549      global mbxG_whichLine
550      global mbxG_subtractLine
551      global mbxG_iconList
552
553      — SCROLL UP WITH HIGHLIGHT
554      — ICONS NOW SCROLL... HOWEVER, ICON SPRITE POSITION IS BASED ON
555      — MBXG_SUBTRACTLINE, NOT ON THE THE ACTUAL CORRESPONDING LINE
556      — NUMBER
557      — OF THE MESSAGE IN THE MAILBOX WINDOW.
558
559      set numberOfIconsVar = count(mbxG_iconList)
560      set lastIconPos = getPropAt(mbxG_iconList, count(mbxG_iconList))
561      if lastIconPos >= 550 then

```

附录A-邮箱显示窗口代码-Page 12

```

562      — there are enough messages to make scrolling nessisary
563      repeat while the mouseDown = TRUE
564
565      —oldSubtractLine gets set to mbxG_subtractLine before mbxG_sub, gets
566      —incremented. This keeps the icons from falling one position behind
567      —it's prospective message
568
569      set oldSubtractLine = mbxG_subtractLine
570      set mbxG_subtractLine = mbxG_subtractLine + 1
571
572      if mbxG_subtractLine > 0 then
573          set mbxG_subtractLine = 0
574      end if
575
576      — SCROLL ALL FIELDS TOGETHER
577
578      scrollByLine member "MessageNumber", -1
579      scrollByLine member "MailboxTo", -1
580      scrollByLine member "MailboxSubject", -1
581      scrollByLine member "MailboxDate", -1
582      scrollByLine member "mime", -1
583      scrollByLine member "Messageread", -1
584
585      set numberIcons = the lineCount of member "MailboxTo" + 40
586      set amountOfMail = the lineCount of member "MailboxTo"
587
588      — for debugging
589      — put "linecount:" & the lineCount of member "MailboxTo"
590      — put "subtractline:" & mbxG_subtractLine
591
592      repeat with i = 40 to numberIcons
593
594
595      — if the following 2 conditions are true, then scroll the icons
596      — i.e. if messages scroll, icons do too, if not, then neither do icons.
597
598      if mbxG_subtractLine >= - amountOfMail + 1 and oldSubtractLine < 0 then
599          set the locV of sprite i = the locV of sprite i + 15
600          if the locV of sprite i < 105 then
601              set the visible of sprite i = FALSE
602          else
603              set the visible of sprite i = TRUE
604          end if
605      else
606          nothing
607      end if
608  end repeat
609
610      — MOVE HIGHLIGHT WITH LINE, MOVING HIGHLIGHT
611      — OFF SCREEN WHEN LINE MOVES OFF SCREEN
612      set whichHighlight = mbxG_whichLine + mbxG_subtractLine

```

附录A-邮箱显示窗口代码-Page 13

```

613 if whichHighlight <= 0 or whichHighlight >= 22 then
614     set the loc of sprite 11 to point (11, -20)
615 else — set the locV of highlight to scrolled message
616     set the locV of Sprite 11 to (99 + linePosToLocV(member "MailboxTo", whichHighlight))
617 end if
618 updateStage
619 end repeat
620 end if
621
622
623 on mouseUp
624     set numberOfIcons = the lineCount of member "MailboxTo" + 40
625     repeat with i = 40 to numberOfIcons
626         if the locV of sprite i > 550 or the locV of sprite i < 105 then
627             set the visible of sprite i = FALSE
628         else
629             set the visible of sprite i = TRUE
630         end if
631     end repeat
632
633
634 on mouseDown
635     global mbxG_whichLine
636     global mbxG_subtractLine
637     global mbxG_iconList
638
639
640 — AS MESSAGES ARE SCROLLED, ICONS NEED TO MOVE WITH THE
641 MESSAGE, AND
642 THE MEMBERNUM
643 — OF ICON SPRITES BE ASSIGNED TO THE NEW MESSAGE THAT IS VISIBLE
644 AFTER
645 — BEING SCROLLED.
646 — ICONS NOW SCROLL... HOWEVER, ICON SPRITE POSITION IS BASED ON
647 — MBXG_SUBTRACTLINE, NOT ON THE THE ACTUAL CORRESPONDING LINE
648 NUMBER
649 — OF THE MESSAGE IN THE MAILBOX WINDOW.
650 set numberOfIconsVar = count(mbxG_iconList)
651 set lastIconPos = getPropAt(mbxG_iconList, count(mbxG_iconList))
652 if lastIconPos >= 550 then
653     — there are enough messages to make scrolling nessisary
654     repeat while the mouseDown = TRUE
655         scrollByLine member "MessageNumber", 1
656         scrollByLine member "MailboxTo", 1
657         scrollByLine member "MailboxSubject", 1
658         scrollByLine member "MailboxDate", 1
659         scrollByLine member "mime", 1
660         scrollByLine member "messageread", 1
661     — GET NUMBER USED TO CORRECT FOR DISCREPANCY
662     — BETWEEN
663     — THE MOUSELINE AND THE LINEPOSTOLOCV

```

附录A-邮箱显示窗口代码-Page 14

```

664      — THE MOUSELINE GIVES LINE WITHIN FIELD TOTAL
665      — THE LINEPOSTOLOCV USES LINE OF FIELD ON SCREEN
666      set mbxG_subtractLine = mbxG_subtractLine - 1
667      set numberOfIcons = the lineCount of member "MailboxTo" + 40
668      set amountOfMail = the lineCount of member "MailboxTo"
669      — put "linecount:" & the lineCount of member "MailboxTo"
670      — put "subtractline:" & mbxG_subtractLine
671      repeat with i = 40 to numberOfIcons
672          if mbxG_subtractLine >= - amountOfMail + 1 then
673              set the locV of sprite i = the locV of sprite i - 15
674              if the locV of sprite i < 105 then
675                  set the visible of sprite i = FALSE
676              else
677                  set the visible of sprite i = TRUE
678              end if
679          else
680              nothing
681          end if
682      end repeat
683
684      if mbxG_subtractLine < - amountOfMail + 1 then
685          set mbxG_subtractLine = - amountOfMail + 1
686      end if
687
688      — MOVE HIGHLIGHT WITH LINE, MOVING HIGHLIGHT
689      — OFF SCREEN WHEN LINE MOVES OFF SCREEN
690
691      set whichHighlight = mbxG_whichLine + mbxG_subtractLine
692      if whichHighlight <= 0 or whichHighlight >= 22 then
693          set the loc of sprite 11 to point (11, -20)
694      else
695          set the locV of Sprite 11 to (99 + linePosToLocV(member
696          "MailboxTo", whichHighlight))
697      end if
698      updateStage
699      end repeat
700  end if
701
702  —
703
704
705  on emc_indicateCheckingInternet
706      global mbxG_red
707      global mbxG_blue
708      global mbxG_white
709      global mbxG_black
710
711      if the locH of sprite 4 > 600 then
712          set the loc of sprite 4 = point(223, 4)
713      end if
714      — if the locH of sprite 5 > 600 then
715          — set the loc of sprite 5 = point(509, 19)
716      — end if

```

附录A-邮箱显示窗口代码-Page 15

```

717     set colorNow = the forecolor of line 1 of member the member of sprite 4
718
719
720     case colorNow of
721         mbxG_black: set colorNext = mbxG_blue — blue
722         mbxG_blue: set colorNext = mbxG_white — pink
723         mbxG_white: set colorNext = mbxG_red — red
724         mbxG_red: set colorNext = mbxG_blue — blue
725     end case
726
727     . set the forecolor of line 1 of member the member of sprite 4 to colorNext
728     updateStage
729
730 end emc_indicateCheckingInternet
731
732 on emc_endIndicateCheckingInternet
733     set the loc of sprite 4 to point(800, 4)
734     set the loc of sprite 5 = point(800, 19)
735     cursor -1
736     updateStage
737 end emc_endIndicateCheckingInternet
738
739 ——— Scripts for the close window button
740
741 on mouseDown
742
743     repeat while the stillDown
744         if inside(point(the mouseH, the mouseV), the rect of sprite the clickon) then
745             if the name of member the member of sprite the clickon = "closeWindow"
746                 then
747                     set the member of sprite the clickon = "closeWindow_down"
748                     updateStage
749                 end if
750             else
751                 set the member of sprite the clickon = "closeWindow"
752                 updateStage
753             end if
754         end repeat
755         set the member of sprite the clickon = "closeWindow"
756         updateStage
757
758 end mouseDown
759
760 on mouseUp
761
762     — Close the window
763     if inside(point(the mouseH, the mouseV), the rect of sprite the clickon) then
764         — these next to lines are to try and speed up
765         — the disposal of the mailbox icons on close
766         — need to check this on slower machine.
767         hideMailIcons(the lineCount of member "MailboxTo")

```

附录A-邮箱显示窗口代码-Page 16

```

768      go frame "stop"
769      tell the stage to emh_killComponent(0, "")
770      set success = the result
771      if success <> TRUE then
772          alert("error closing mailbox MIAW")
773      end if
774  end if
775 end
776
777 on hideMailIcons numberOfIcons
778     repeat with i = 40 to (40 + numberOfIcons)
779         set the visible of sprite i = FALSE
780     end repeat
781     updateStage
782 end
783
784 — script for the open button
785
786 on mouseDown
787
788     repeat while the stillDown
789         if inside(point(the mouseH, the mouseV), the rect of sprite the clickon) then
790             if the name of member the member of sprite the clickon = "open" then
791                 set the member of sprite the clickon = "open_down"
792                 updateStage
793             end if
794         else
795             set the member of sprite the clickon = "open"
796             updateStage
797         end if
798     end repeat
799     set the member of sprite the clickon = "open"
800     updateStage
801 end mouseDown
802
803
804

```

附录A-邮箱显示窗口代码-Page 17

```
805 on mouseUp
806
807     if inside(point(the mouseH, the mouseV), the rect of sprite the clickon) then
808         global mbxG_whichLine, mbxG_messages
809         if mbxG_whichLine = 0 then
810             alert "Select a message by clicking with your mouse."
811             exit
812         end if
813         set maildata = getAt(mbxG_messages, mbxG_whichLine)
814         tell the stage
815             emh_openMessage(maildata)
816         end tell
817     end if
818 end
```

附录 B -类型更新器代码 – Page 1

```

1 --- TYPE_UPDATER IMPLEMENTATION
2 --- Code for a component that maintains message type information for the
3 --- KidCode electronic mail client.
4 --- This Director MIAW makes public functions available for calling by other components
5 --- of KidCode
6
7 --- public functions
8 --- 1. Initialize_TypeTable
9 --- 2. Install_Type
10 --- 3. Uninstall_Type
11
12
13
14 ----- private functions for internal use only
15 ----- 1. Write_Typetable_File
16 ----- 2. Read_Typetable_File
17 ----- 3. Read_Icon_Files_To_RAM
18 ----- 4. read_iconFile
19 ----- 5. delete_mimetype
20 ----- 6. insert_mimetype
21 ----- 7. delete_filetype
22 ----- 8. insert_filetype
23
24
25 --- Filename for permanent storage version of TYPETABLE file
26 --- typetable.txt --default directory is the currentPath directory
27
28
29 --- INITIALIZE_TYPETABLE initializes the data structures used to lookup Mime type
30 icons, attachment filetype icons and message handler MIAWs.
31 --- The SG_Typetable is set up prior to its use. File attachment information is not looked
32 up until it is used when a message with an attached file is encountered.
33
34 on Initialize_TypeTable
35 global SG_TYPETABLE --- super global variable shared across different MIAWS for
36 Typetable
37 global SG_ATTACH_TYPETABLE ---- maintains info for attachment filetypes
38
39 set SG_TYPETABLE = [:] -- initialize property list for mimetype information
40
41 -- initialize property list for filetype information
42 -- this list will be filled only as messages with attachments are encountered
43 set SG_ATTACH_TYPETABLE = [:]
44 set SG_TYPETABLE = Read_TypeTable_File(the pathname & "typetable.txt")
45
46 if count(SG_TYPETABLE) = 0 then -- failed to read typetable file
47   alert("Error: Failed to read the file of MIMEtypes")
48   return (0)
49 end if
50
51 set retVal = Read_Icon_Files_To_RAM()

```

附录 B -类型更新器代码 – Page 2

```

52     if retVal = 0 then
53         alert("Error: Failed to load MIME type icons.")
54         return (0)
55     else
56         return(1)
57     end if
58
59
60 end --- Initialize_TypeTable
61
62
63
64 --- INSTALL_TYPE is used to install a new MIME type into the system.
65 INSTALL_TYPE takes as input a mimetyp (string), a filename of the message handling
66 movie, an filename of the bitmap that contains the mimetyp icon and, optionally, a file
67 extension (string). The function adds the information associated with the MIME type
68 (given by the function parameters) into the MIME type table recorded in permanent storage.
69 Here we use the file "typetable.txt" for permanent storage of the MIME type info.
70
71 on Install_Type mimeToInstall, msgHandler_filename, icon_filename, filetype
72 global SG_TYPETABLE --- information on all installed MIME types
73
74 set DEFAULT_ICONFILE = "defaultIcon.bmp"
75 set SG_TYPETABLE = [:] -- initialize property list for mimetyp information
76
77 ---read the existing MIME type information into RAM
78 set SG_TYPETABLE = Read_TypeTable_File(the pathname & " typetable.txt")
79
80 if count(SG_TYPETABLE) = 0 then -- failed to read typetable file
81     alert ("Error: Failed to read the file of MIMEmes")
82     return (0)
83 end if
84
85 ---Check to see if mimetyp is already installed
86
87 set mimeProperties = get_mimetyp(mimeToInstall)
88
89 if mimeProperties <> 0 then ---mimetyp is already installed .
90     set redefineAlert = baMsgBox(theMessage, "KidCode", "YesNoCancel", "Question", 1)
91
92 -- the alert function should not save the message, only do the alert
93 case redefineAlert of
94     "No": return 0
95     "Cancel": return nil
96     otherwise: nothing -- continue
97 end case
98 end if
99
100 --- Define the new mimetyp
101
102 if verifyMessageHandler(msgHandler_filename) = 0 then --something wrong with

```

附录 B -类型更新器代码 – Page 3

```

103 program file
104     alert("Error: invalid message handler program" && msgHandler_filename)
105     return(0)
106 end if
107
108 insert_msgHandler(SG_TYPETABLE, mimeToInstall, msgHandler_filename)
109
110 if verifyIconImage(icon_filename) = 0 then --something wrong with icon file
111     alert("Error: invalid icon file" && msgHandler_filename ". Using default icon.")
112     insert_iconFileName(SG_TYPETABLE, mimeToInstall, DEFAULT_ICONFILE)
113 else
114     insert_iconFileName(SG_TYPETABLE, mimeToInstall, icon_filename)
115 end if
116
117 if filetype <>"" then
118     insert_filetype(mimeToInstall, filetype)
119     writeTypeToRegistry(mimeToInstall, filetype)
120 end if
121
122 set retVal = write_TypeTable_File()
123 if retVal = 0 then
124     alert("Error writing typetable to file." && mimeToInstall && "not installed.")
125     return(0)
126 else return(1)
127
128 end -- Install_Type
129
130
131 ----- UNINSTALL_TYPE removes a mimetype and its properties from both the file and
132 the global variable SG_TYPETABLE
133
134 on unInstall_Type mimeType
135 global SG_TYPETABLE -- information on all installed MIME types
136
137 set SG_TYPETABLE = [:] -- initialize property list for mimetype information
138
139 --read the existing MIME type information into RAM
140 set SG_TYPETABLE = Read_TypeTable_File(the pathname & " typetable.txt")
141
142 if count(SG_TYPETABLE) = 0 then -- failed to read typetable file
143     alert ("Error: Failed to read the file of MIMETypes")
144     return (0)
145 end if
146
147 delete_mimetype(mimeType)
148
149 -- write the revised typetable to the file
150 set retVal = write_TypeTable_File()
151 if retVal = 0 then
152     alert("Error." && mimeToInstall && "could not be uninstalled. Typetable file write
error.")
153

```

 附录 B -类型更新器代码 – Page 4

```

154      return(0)
155  else return(1)
156
157  end unInstall_Type
158
159  --- WRITE_TYPETABLE_FILE writes the information in SG_TYPETABLE to the
160  typetable file on disk. This file stores properties associated with each mimetype.
161  ---- SG_TYPETABLE is a property list that contains a list of mimetypes.
162
163  on Write_Typetable_File
164  global SG_TYPETABLE    --- super global variable shared across different MIAWS
165  set fileName = the pathname & "typetable.txt"
166  set bkupFileName = the pathname & "typetable.bak"
167
168  if count(SG_TYPETABLE) = 0 then   -- no mimetypes defined
169      alert("Error. No mimetype data to write.")
170  return(0)
171  end if
172
173  --- create backup for typetable file
174  copyFile(fileName, bkupFileName)
175
176  -- start up Fileio Xtra
177  set mFile = new(xtra "fileio")
178
179  set retVal = deleteFile(mFile, filename)  -- delete old version before rewriting
180  set retVal = createFile(mFile, filename)
181  if retVal = 0 then
182      alert("Error updating typetable file.")
183      renameFile(bkupFileName, filename)
184      return(0)
185  end if
186
187  openFile(mFile, fileName, 2) -- open for write access
188  setPosition(mFile, 0)
189
190  --- write the data into the file
191  set i=1
192  set mimeType = getA1(SG_TYPETABLE, i)
193  repeat while mimeType <> 0
194      set dataToWrite = mimeType
195      put " " & get_filetype(mimeType) into dataToWrite
196      put " " & get_iconFileName(mimeType) into dataToWrite
197      put " " & get_MsgHandler(mimeType) into dataToWrite
198      writeln(mFile, dataToWrite)
199  set i= i+1
200  set mimeType = getA1(SG_TYPETABLE, i)
201  end repeat
202
203  closeFile(mFile)
204  set retVal = deleteFile(mFile, bkupFileName)  -- delete backup file

```

附录 B -类型更新器代码 – Page 5

```

205    return(0)
206  end  Write_TypeTable_File
207  --- READ_TYPETABLE_FILE reads the typetable file
208  ---- and creates a data structure in memory, SG_TYPETABLE
209  ---- SG_TYPETABLE is a property list that contains a list of mimetypes
210
211 When the function returns, the global property list data structure, SG_TYPETABLE,
212 contains an entry for each mimetype. Along with the pathName for the message handling
213 movie and the IconFile. Later the cast member number for the icon in RAM will be added
214 to the datastructure. For now these are all set to 0. This data structure looks like,
215 [ "text/plain": ["txt", 0, "C:\KidCode\text.gif", "C:\KidCode\text.dxr"], "x-
216 application/grid": [ "", 0, "C:\KidCode\grid.gif", "C:\KidCode\grid.dxr"] ]
217
218 on Read_TypeTable_File
219 global SG_TYPETABLE      --- super global variable shared across different MIAWS
220 set fileName = the pathname & "typetable.txt"
221
222 set SG_TYPETABLE = [:] -- initialize property list for mimetypes
223
224 -- start up Fileio Xtra
225 set mFile = new(xtra "fileio")
226 openFile(mFile, fileName, 1) -- open for read only access
227 set status = status(mFile)
228
229 if status <> 0 then
230   alert("Error. Could not open mimetype table: " & error(mFile, status))
231   closeFile(mFile) -- just to be safe
232   return FALSE
233 end if
234
235 setPosition(mFile, 0)
236
237 -- Lingo can't read one line at a time so simulate this by reading the entire file into the
238 string, str
239 set str = readFile(mFile)
240
241 set nTypes = the number of lines in str
242 repeat with j = 1 to nTypes
243   set mimetype = word 1 of line j
244   insert_mimetype(mimetype)
245   insert_filetype(mimetype, word 2 of line j)
246   insert_iconFileName(mimetype, word 3 of line j)
247   insert_msgHandler(mimetype, word 4 of line j)
248 end repeat
249
250 closeFile(mFile)
251
252 end  Read_TypeTable_File
253
254 on Read_Icon_Files_To_RAM

```

附录 B -类型更新器代码 – Page 6

```

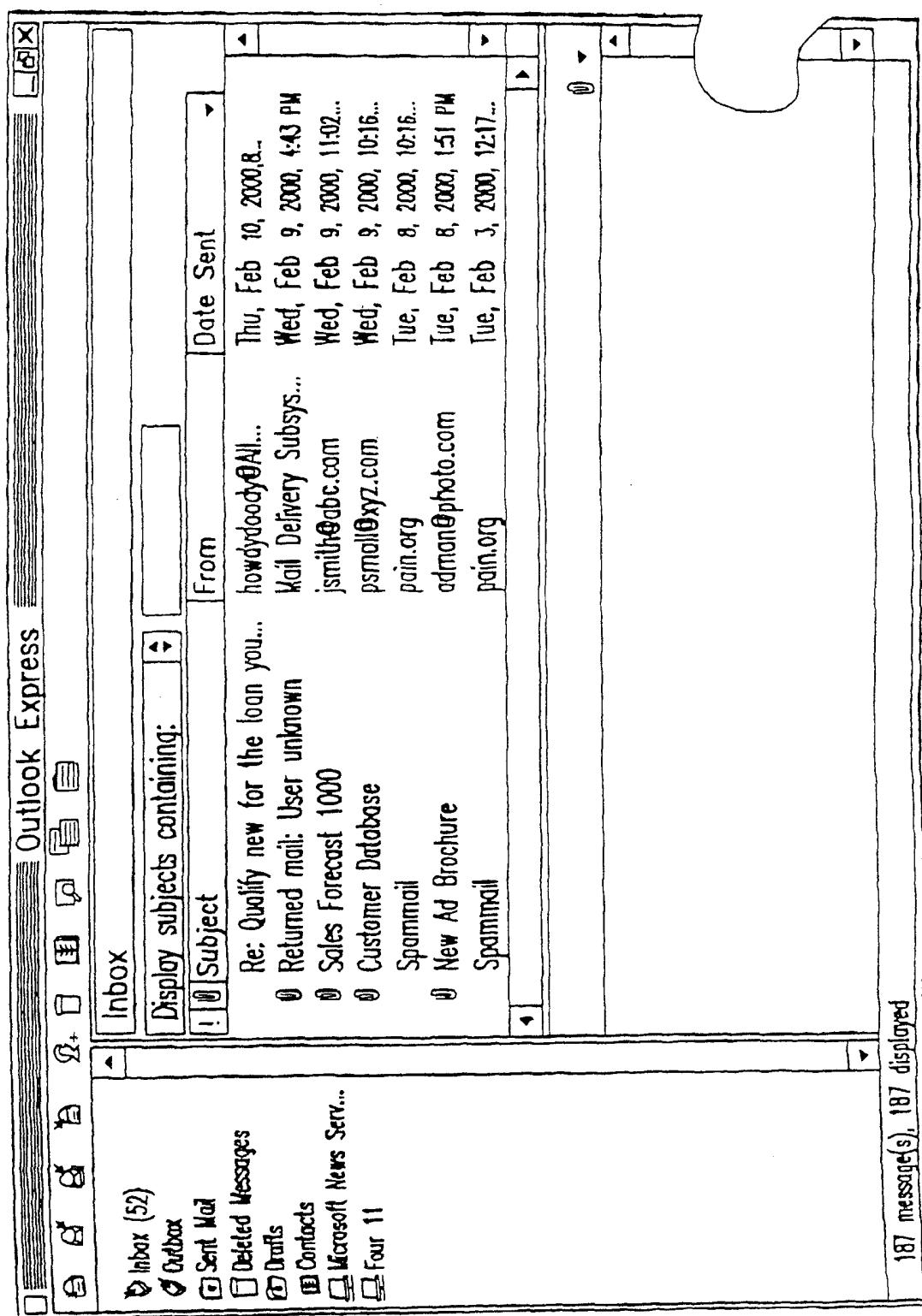
256     global SG_TYPETABLE
257     global SG_DEFAULT_ICON_PTR = 1000
258
259     if count(SG_TYPETABLE) = 0 then -- no mimetypes defined
260         alert("Error. No mimetype data. Can't load icons.")
261     return(0)
262 end if
263
264 ---load the default icon
265 importFileInto(member SG_DEFAULT_ICON_PTR, the pathname & "defaulticon.gif")
266
267 --- .cycle through the mimetypes loading icons
268 set castNum = SG_DEFAULT_ICON_PTR + 1 --first icon immediately follows the
269 default
270 set i= 1
271 set mimeType = getAt(SG_TYPETABLE, i)
272 repeat while mimeType <> 0
273     set iconFile = getFileFromName(SG_TYPETABLE, mimeType)
274     if icon = "" then --- icon not defined use default
275         set iconPtr = SG_DEFAULT_ICON_PTR
276     else
277         set iconPtr = read_IconFile(iconFile, castNum )
278     if iconPtr > 0 then
279         set castNum = castNum + 1
280     else set iconPtr = SG_DEFAULT_ICON_PTR
281     end if
282
283     insert_iconPtr(mimeType, iconPtr)
284 set i= i+1
285 set mimeType = getAt(SG_TYPETABLE, i)
286 end repeat
287
288 return(1)
289
290 end Read_Icon_Files_To_RAM
291
292
293
294 --- READ_ICONFILE loads a single icon bitmap into RAM
295 on read_IconFile filename, castMemberNum
296 set retVal = importFileInto(member castNum, iconFile)
297 return (retVal)
298 end read_IconFile
299 --- Data Access Functions for MIMETYPE info
300 --- Data is stored in the property list SG_TYPETABLE with the following structure:
301 --- mimetype: properties
302 --- where mimetype is a string, e.g. "text/plain"
303 --- and properties is a list with the following elements:
304     [ filetype, iconPtr, iconFileName, msgHandler_FileName
305     --- e.g. SG_TYPETABLE =
306     --- [ "text/plain": ["txt", 0, "C:\KidCode\text.gif", "C:\KidCode\text.dxr"],
```

附录 B -类型更新器代码 – Page 7

```
307     ---- "x-application/grid": [ "", 0, "C:\KidCode\grid.gif", "C:\KidCode\grid.dxr" ]]
308
309     on get_mimetype mimetype
310         global SG_TYPETABLE
311         return( getProp(SG_TYPETABLE, mimetype) )
312     end
313
314     on get_filetype mimetype
315         global SG_TYPETABLE
316         set theProperties = getProp(SG_TYPETABLE, mimetype)
317         return( getAt(theProperties, 1) )
318     end
319
320     on get_iconPtr mimetype
321         global SG_TYPETABLE
322         set theProperties = getProp(SG_TYPETABLE, mimetype)
323         return( getAt(theProperties, 2) )
324     end
325
326     on get_iconFileName mimetype
327         global SG_TYPETABLE
328         set theProperties = getProp(SG_TYPETABLE, mimetype)
329         return( getAt(theProperties, 3) )
330     end
331
332     on get_msgHandler mimetype
333         global SG_TYPETABLE
334         set theProperties = getProp(SG_TYPETABLE, mimetype)
335         return( getAt(theProperties, 4) )
336     end
337
338     on insert_mimetype mimetype
339         global SG_TYPETABLE
340         addProp(SG_TYPETABLE, mimetype)
341     end
342
343     on insert_filetype mimetype, filetype
344         global SG_TYPETABLE
345         set theProperties = getProp(SG_TYPETABLE, mimetype)
346             add(theProperties, filetype)
347             setProp(SG_TYPETABLE, mimetype, theProperties)
348     end
349
350     on insert_iconPtr mimetype, iconPtr
351         global SG_TYPETABLE
352         set theProperties = getProp(SG_TYPETABLE, mimetype)
353             add(theProperties, iconPtr)
354             setProp(SG_TYPETABLE, mimetype, theProperties)
355     end
356
357     on insert_iconFileName mimetype, iconfilename
```

附录 B -类型更新器代码 – Page 8

```
358     global SG_TYPETABLE
359     set theProperties = getProp(SG_TYPETABLE, mimetype)
360         add(theProperties, iconfilename)
361         setProp(SG_TYPETABLE, mimetype, theProperties)
362     end
363
364     on insert_msgHandler mimetype
365     global SG_TYPETABLE
366     set theProperties = getProp(SG_TYPETABLE, mimetype)
367         add(theProperties, msgHandler)
368         setProp(SG_TYPETABLE, mimetype, theProperties)
369     end
370
371
372     on delete_mimetype mimetype
373     global SG_TYPETABLE
374     deleteProp(SG_TYPETABLE, mimetype)
375     end
376
377     on delete_filetype mimetype
378     global SG_TYPETABLE
379     set properties = getProp(SG_TYPETABLE, mimetype)
380         setAt(properties, 1, "")
381         setProp(SG_TYPETABLE, mimetype, properties)
382     end
383
384     on delete_icon mimetype
385     global SG_TYPETABLE
386     set properties = getProp(SG_TYPETABLE, mimetype)
387         setAt(properties, 2, 0)
388         setProp(SG_TYPETABLE, mimetype, properties)
389     end
390
391     on delete_iconFileName mimetype
392     global SG_TYPETABLE
393     set properties = getProp(SG_TYPETABLE, mimetype)
394         setAt(properties, 3, "")
395         setProp(SG_TYPETABLE, mimetype, properties)
396     end
397
398     on delete_msgHandler mimetype
399     global SG_TYPETABLE
400     set properties = getProp(SG_TYPETABLE, mimetype)
401         setAt(properties, 4, "")
402         setProp(SG_TYPETABLE, mimetype, properties)
403     end
```



现有技术
图 1

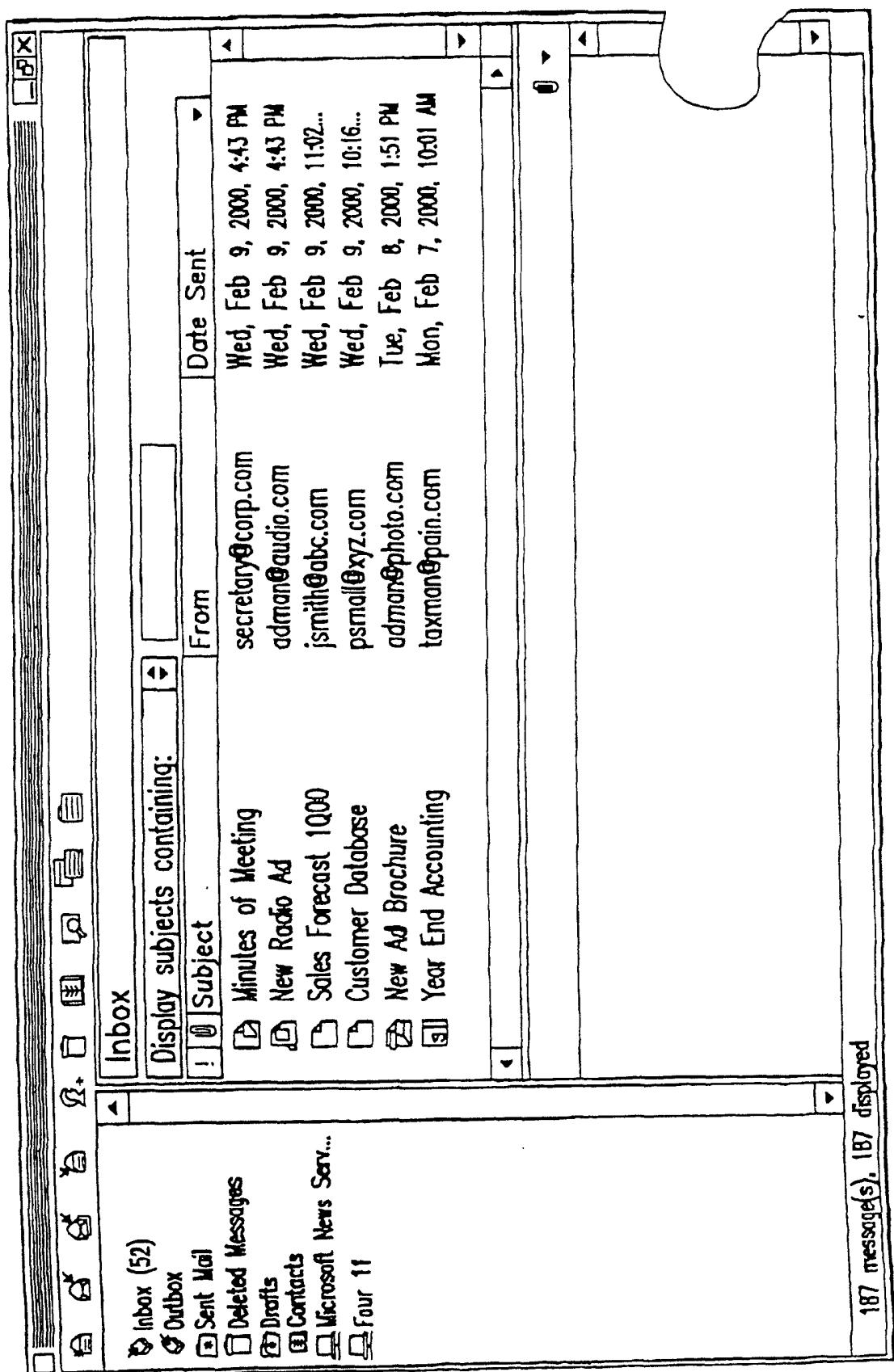
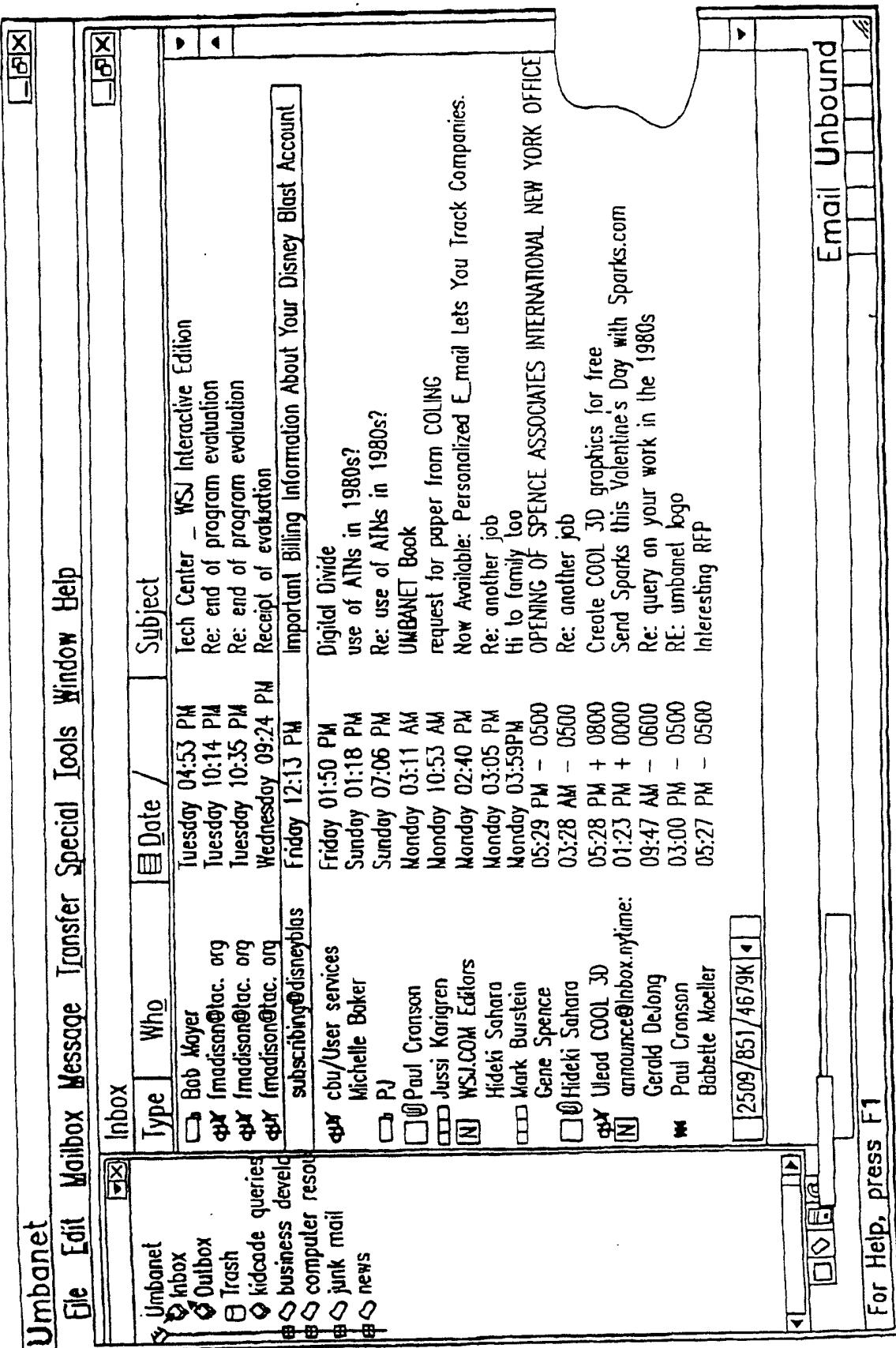


图 2



2a

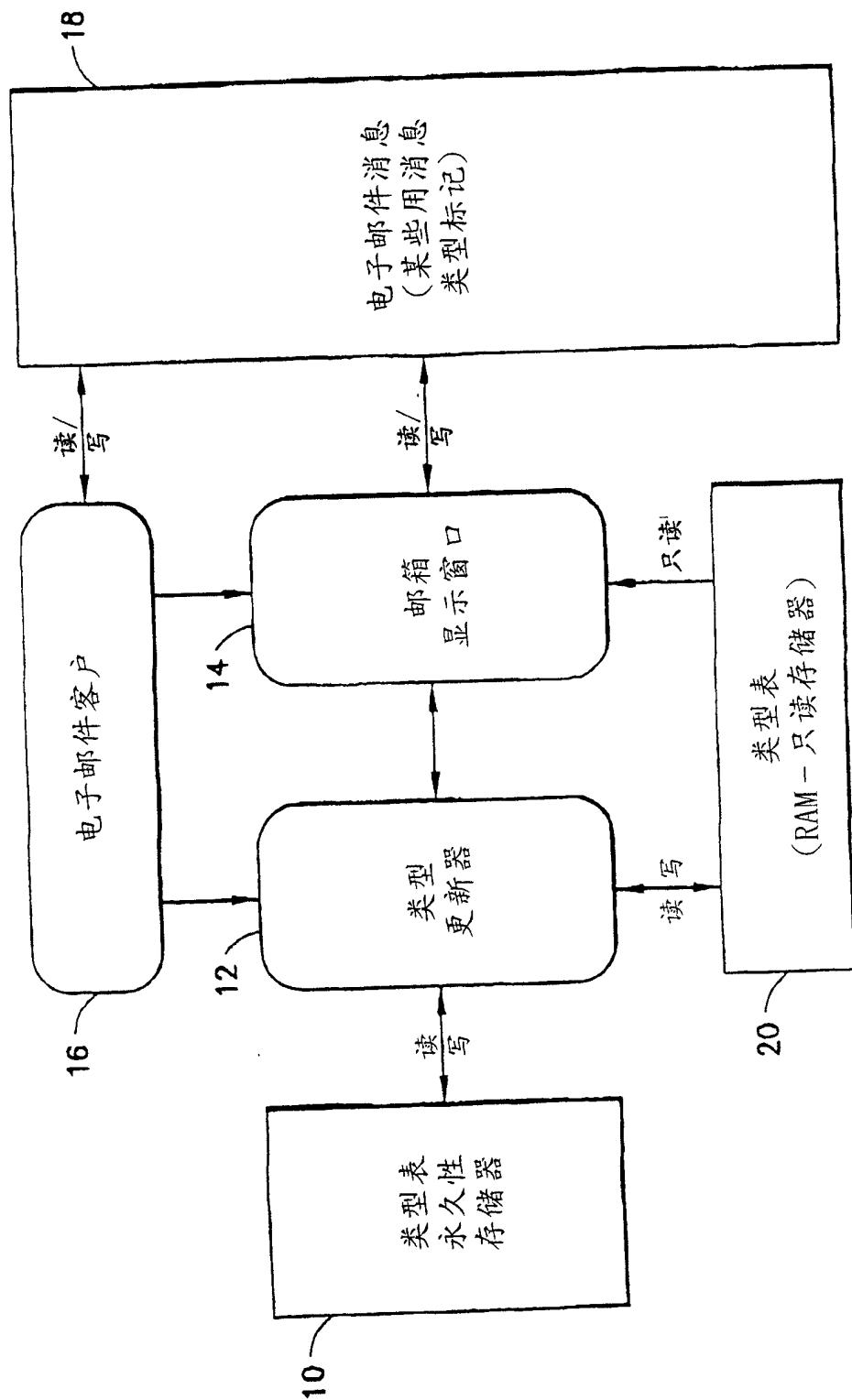


图 3

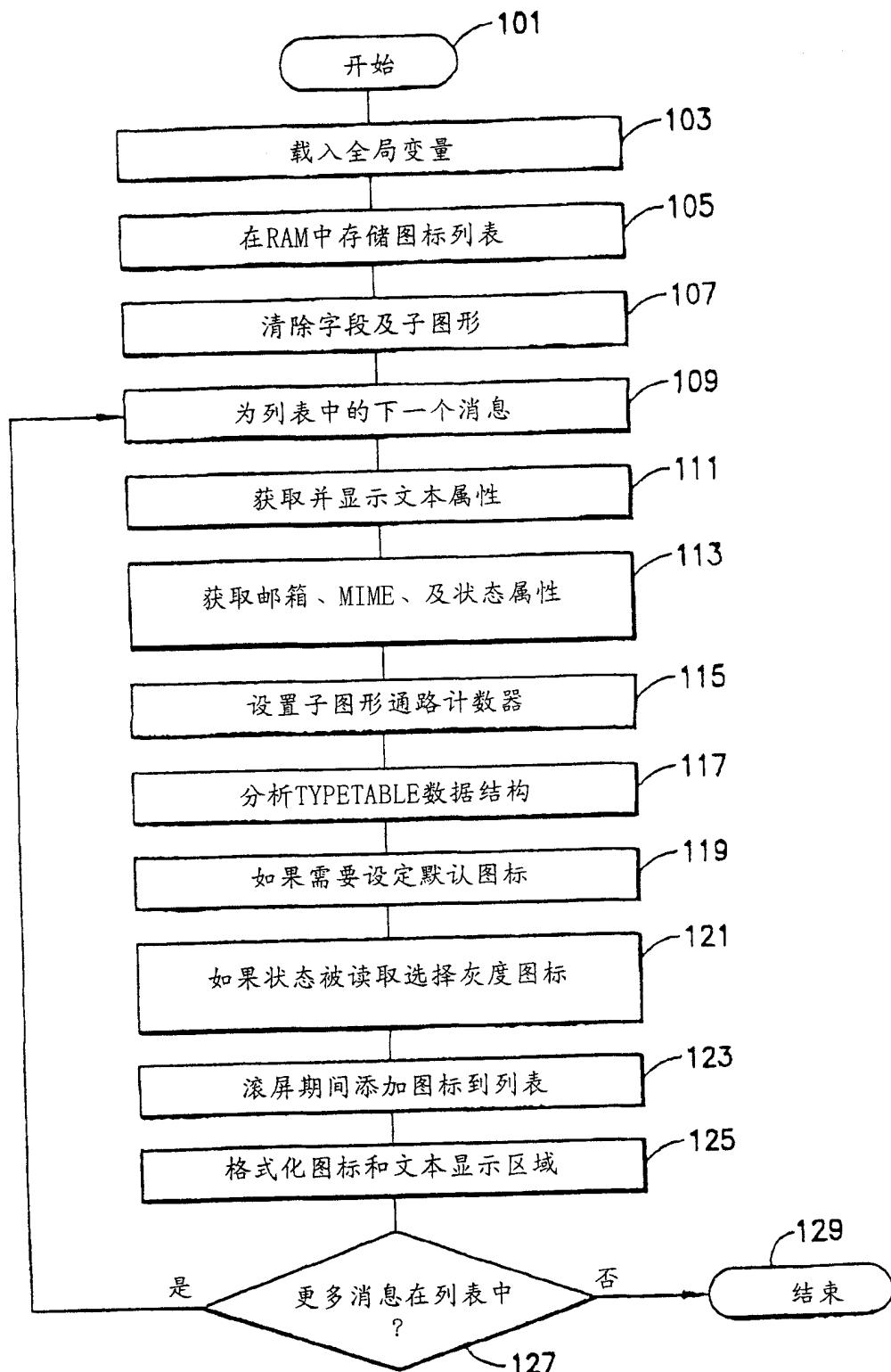


图 4

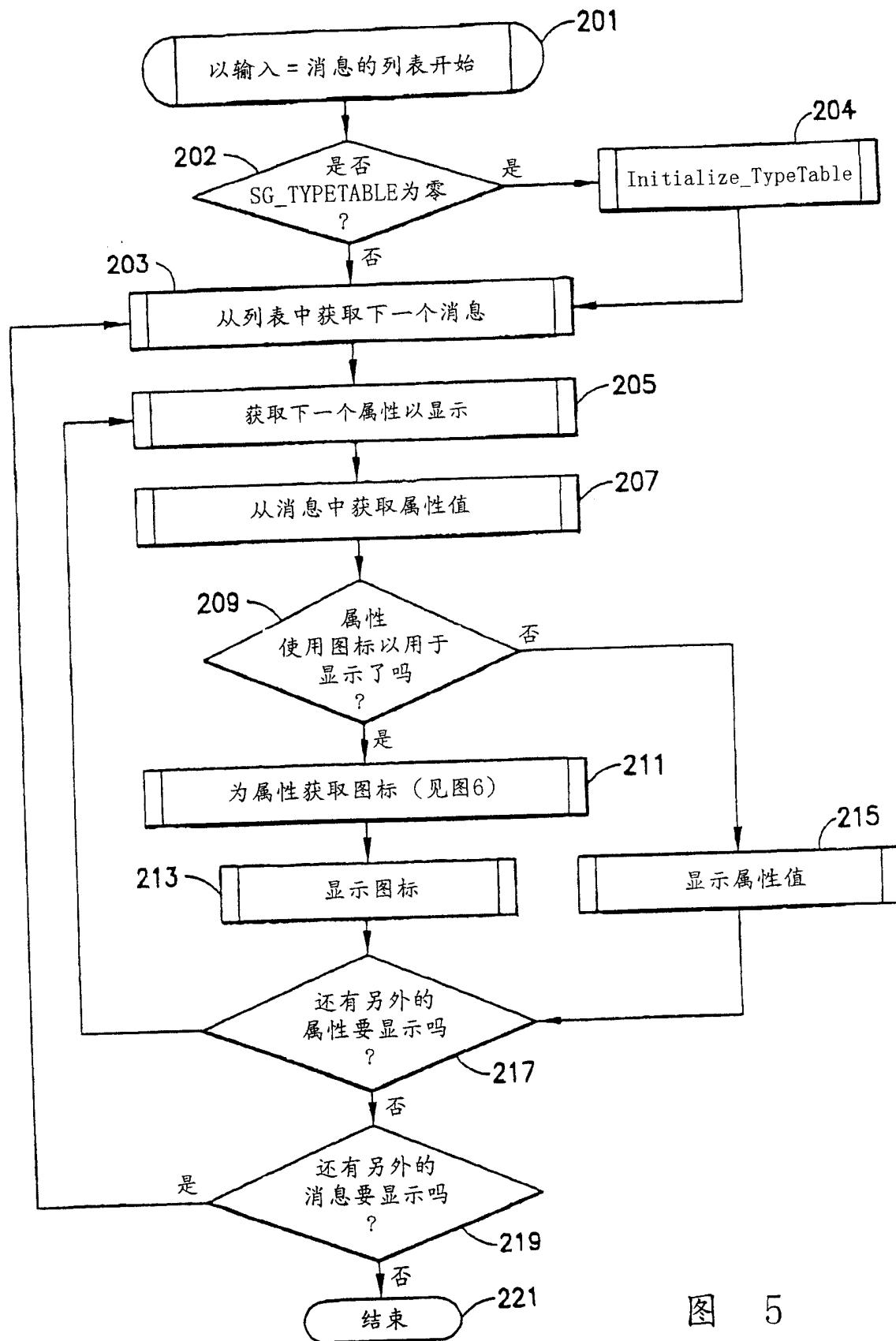


图 5

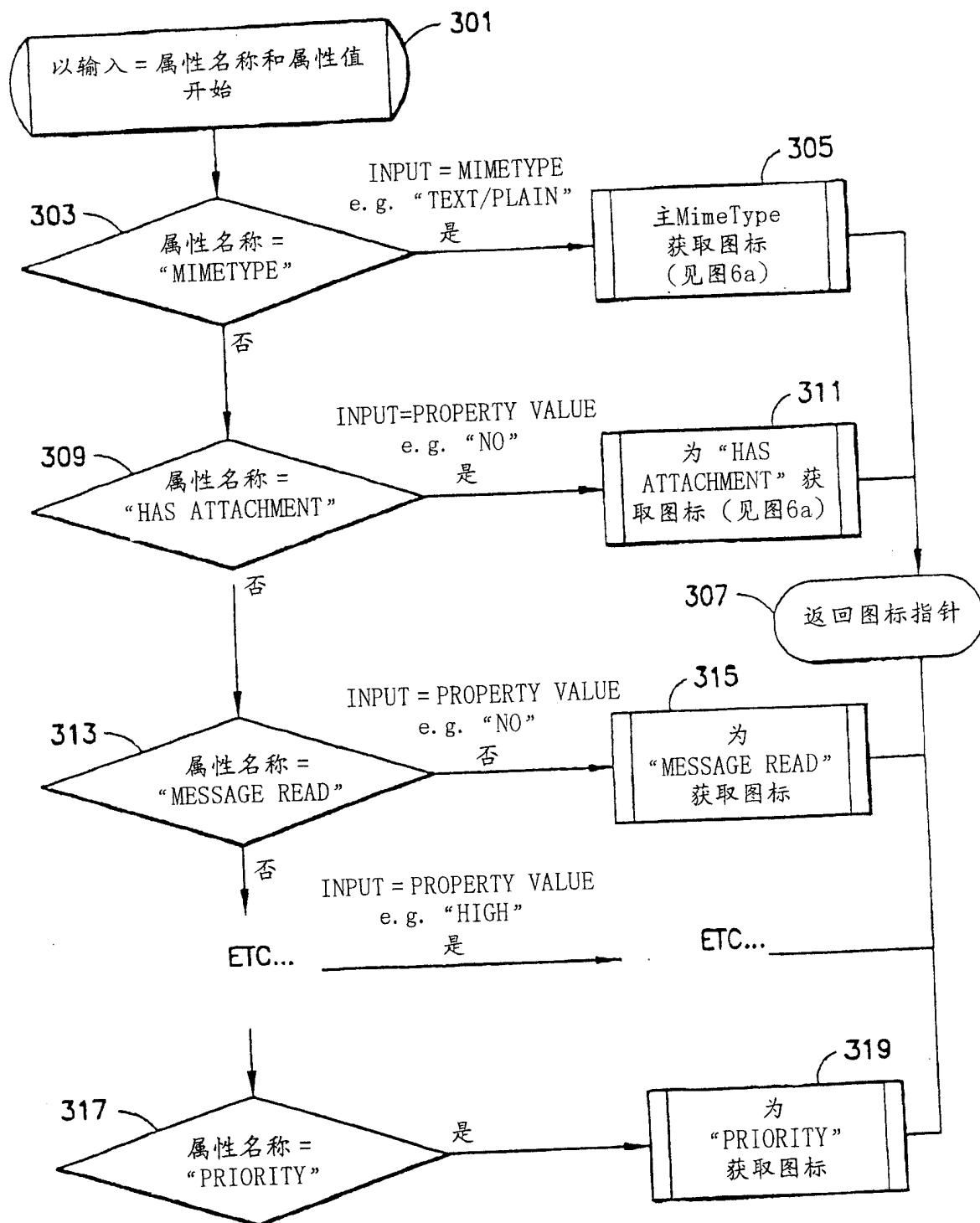


图 6

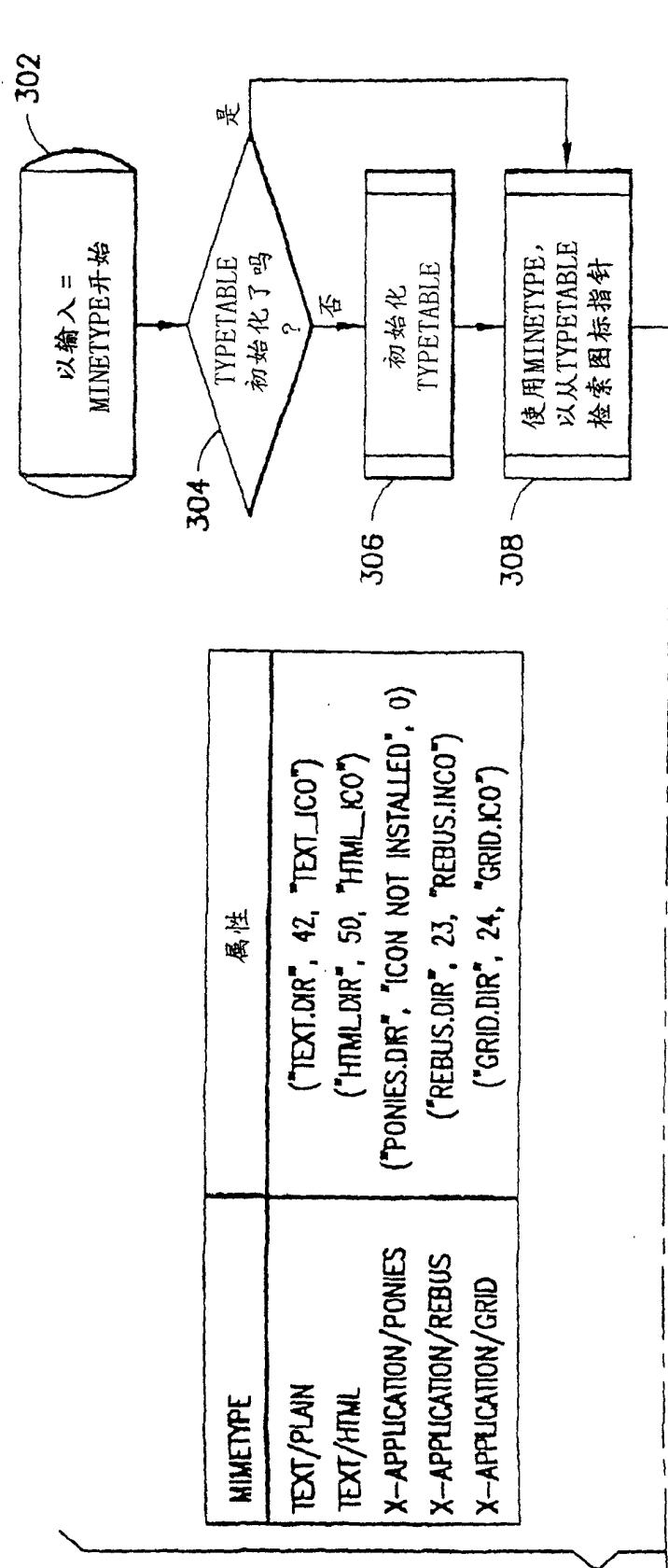


图 6a-1

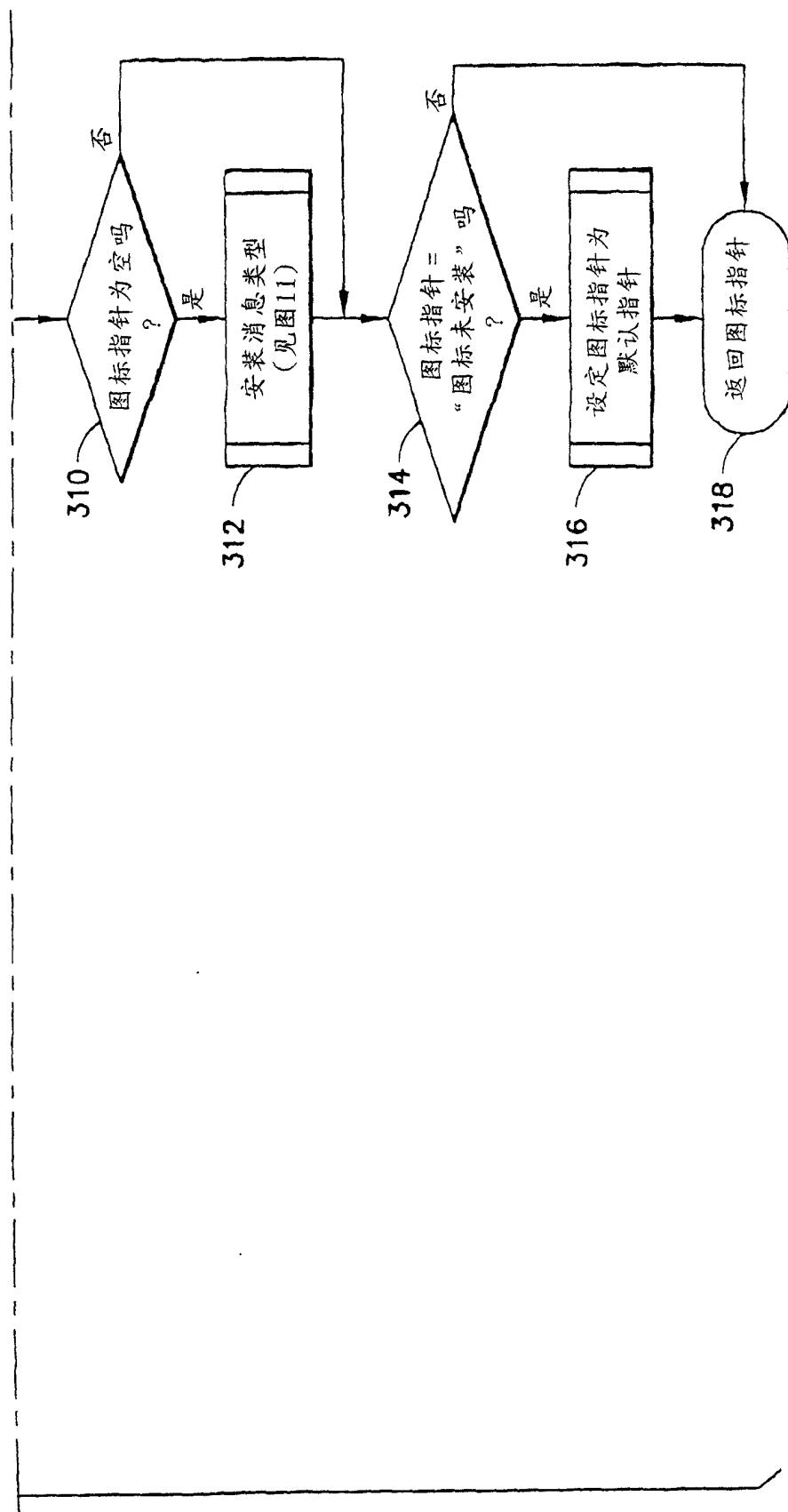


图 6a-2

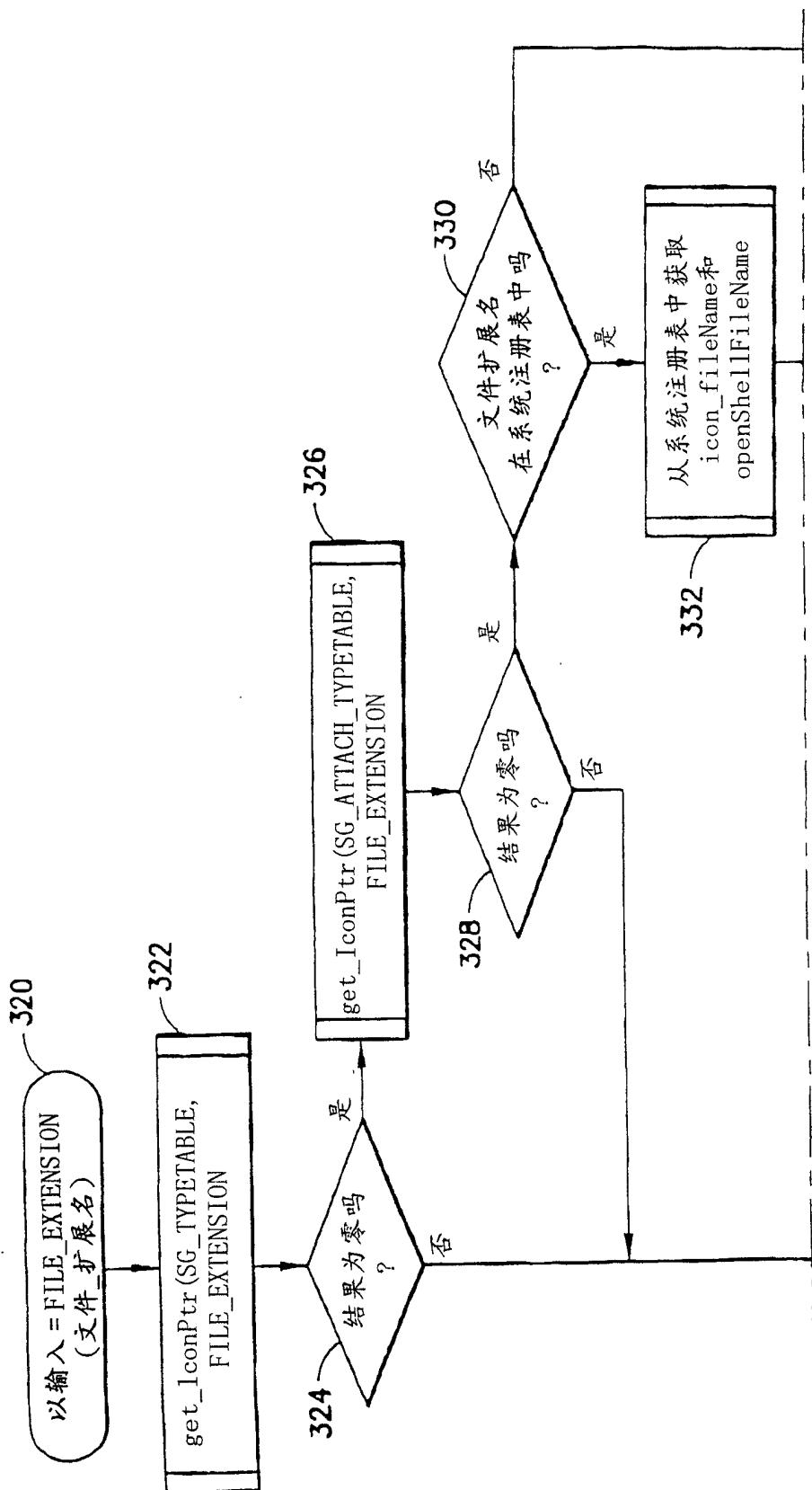


图 6b-1

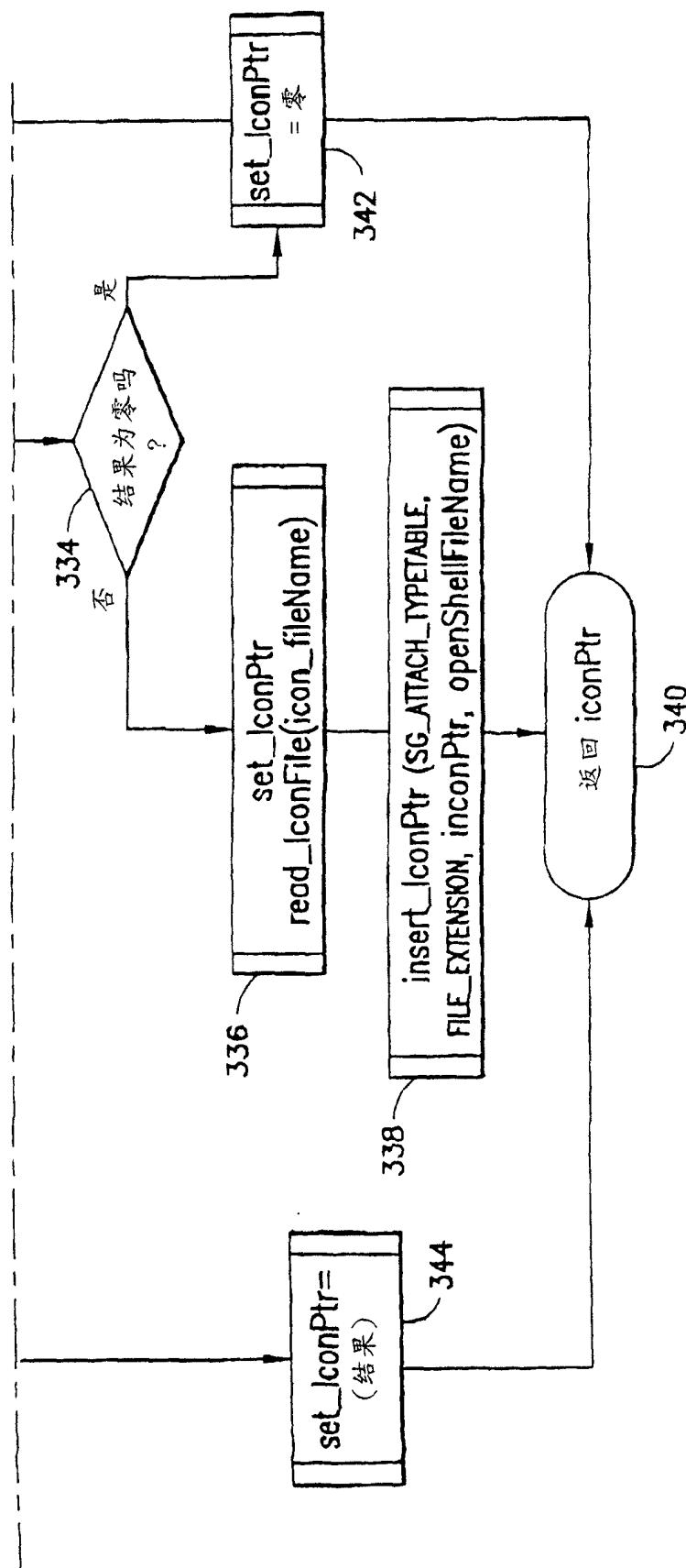


图 6b-2

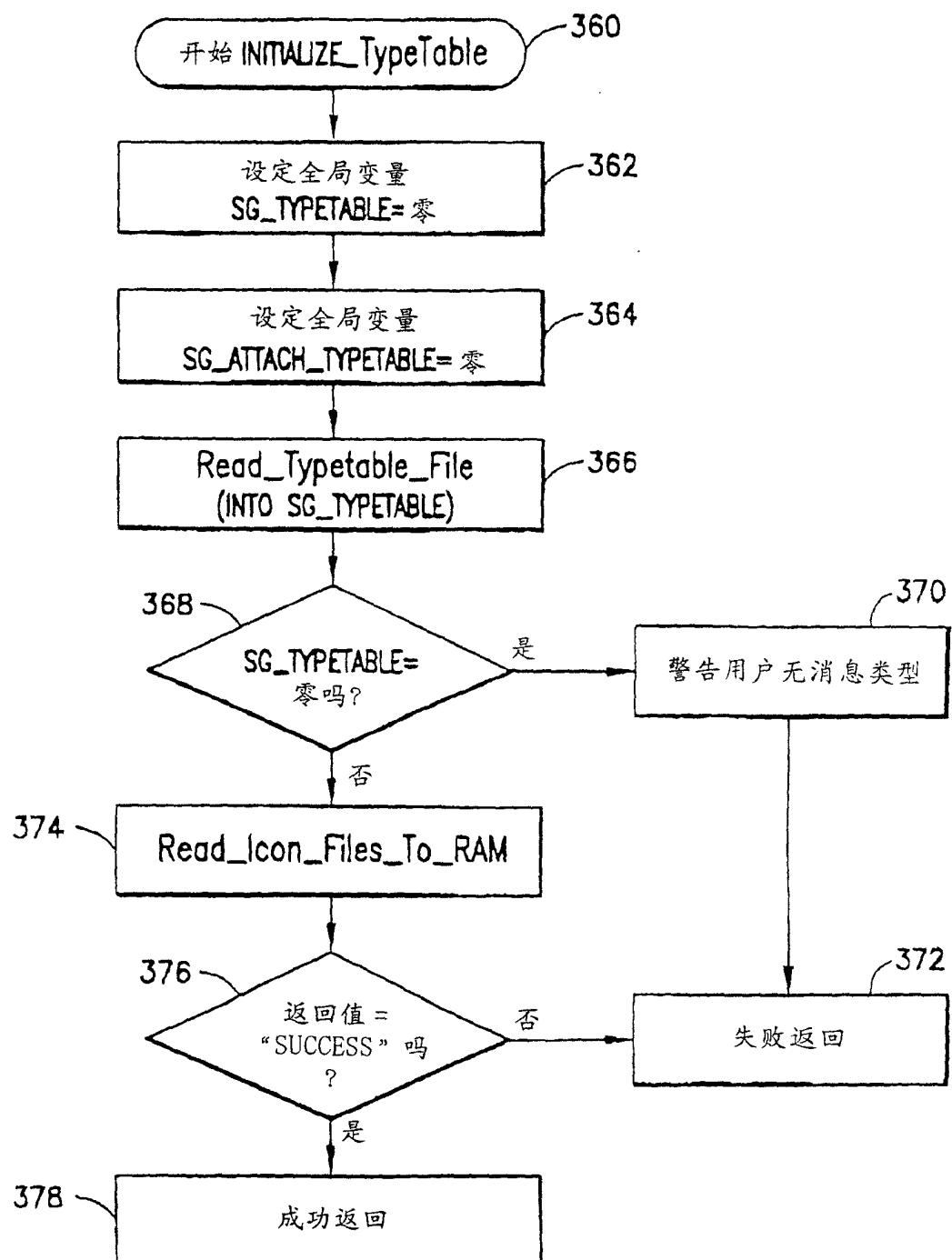


图 7

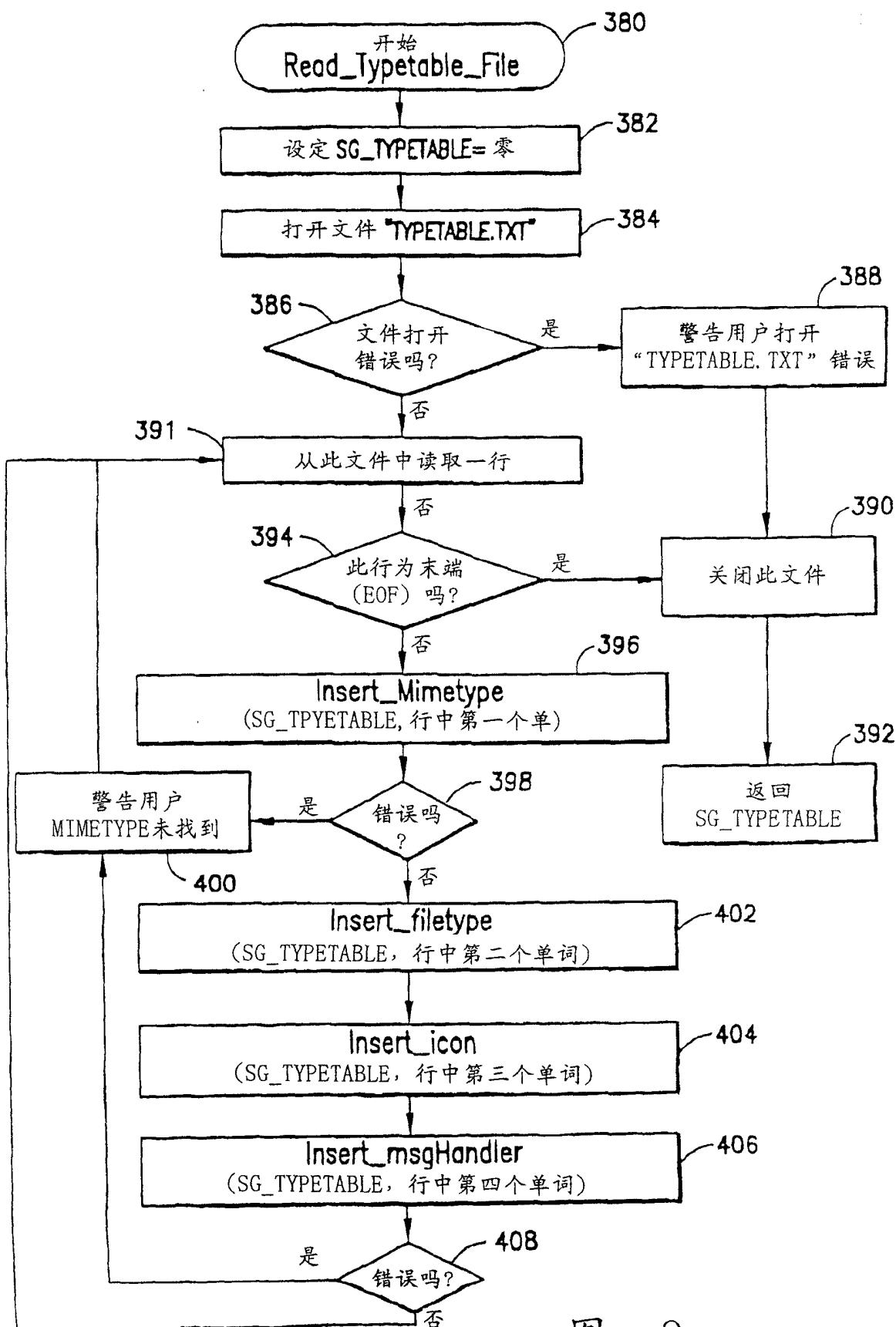
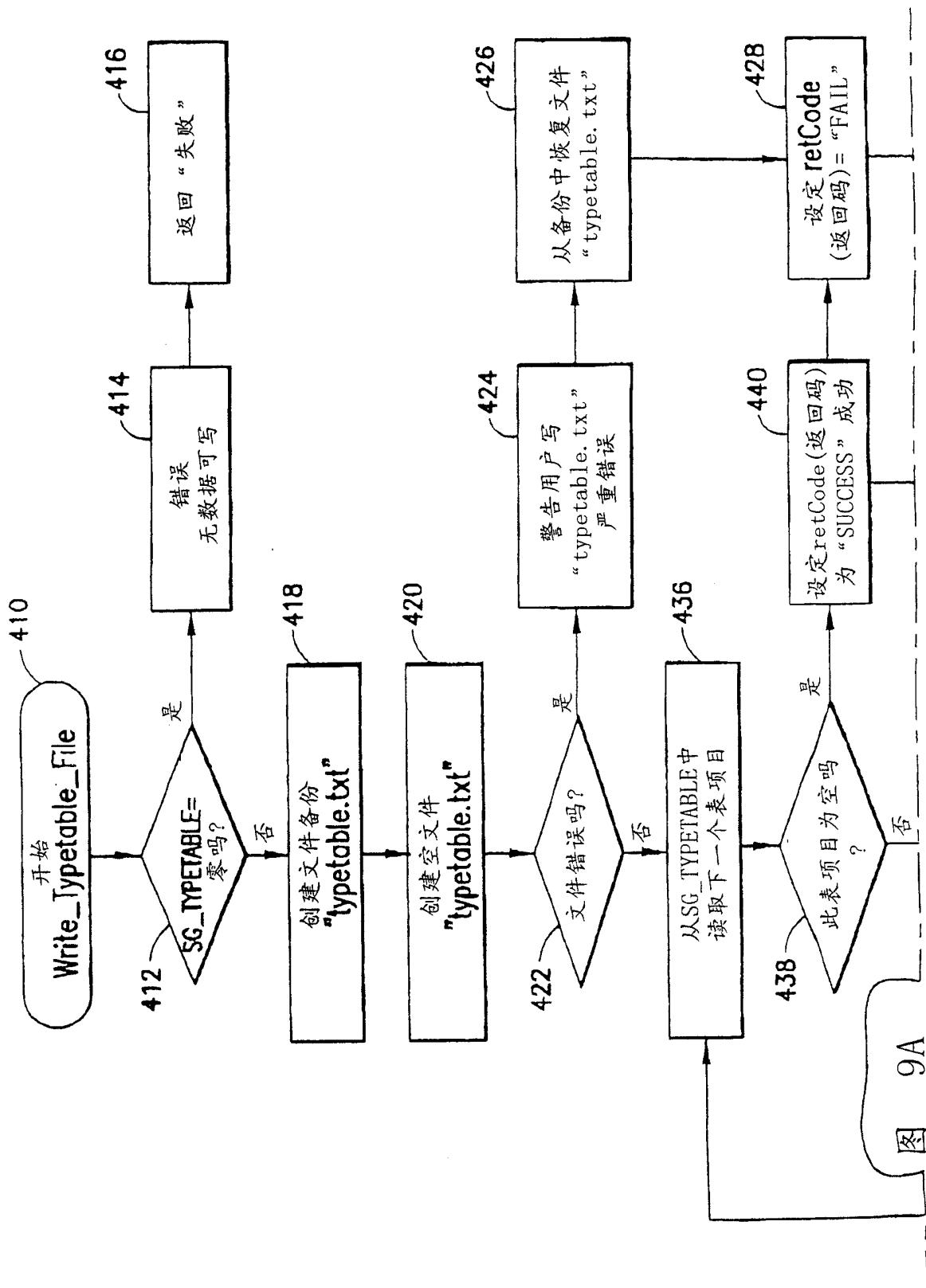
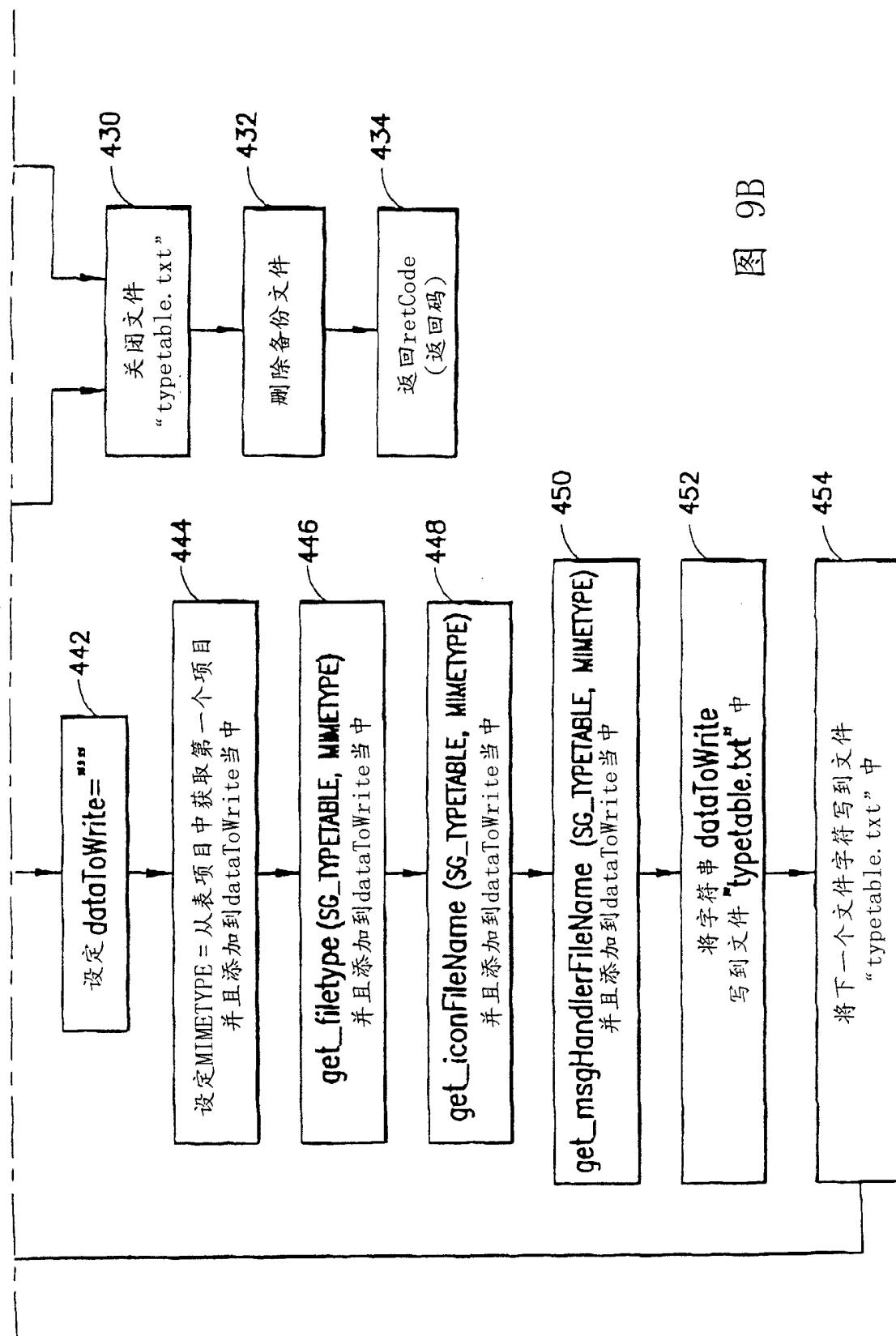
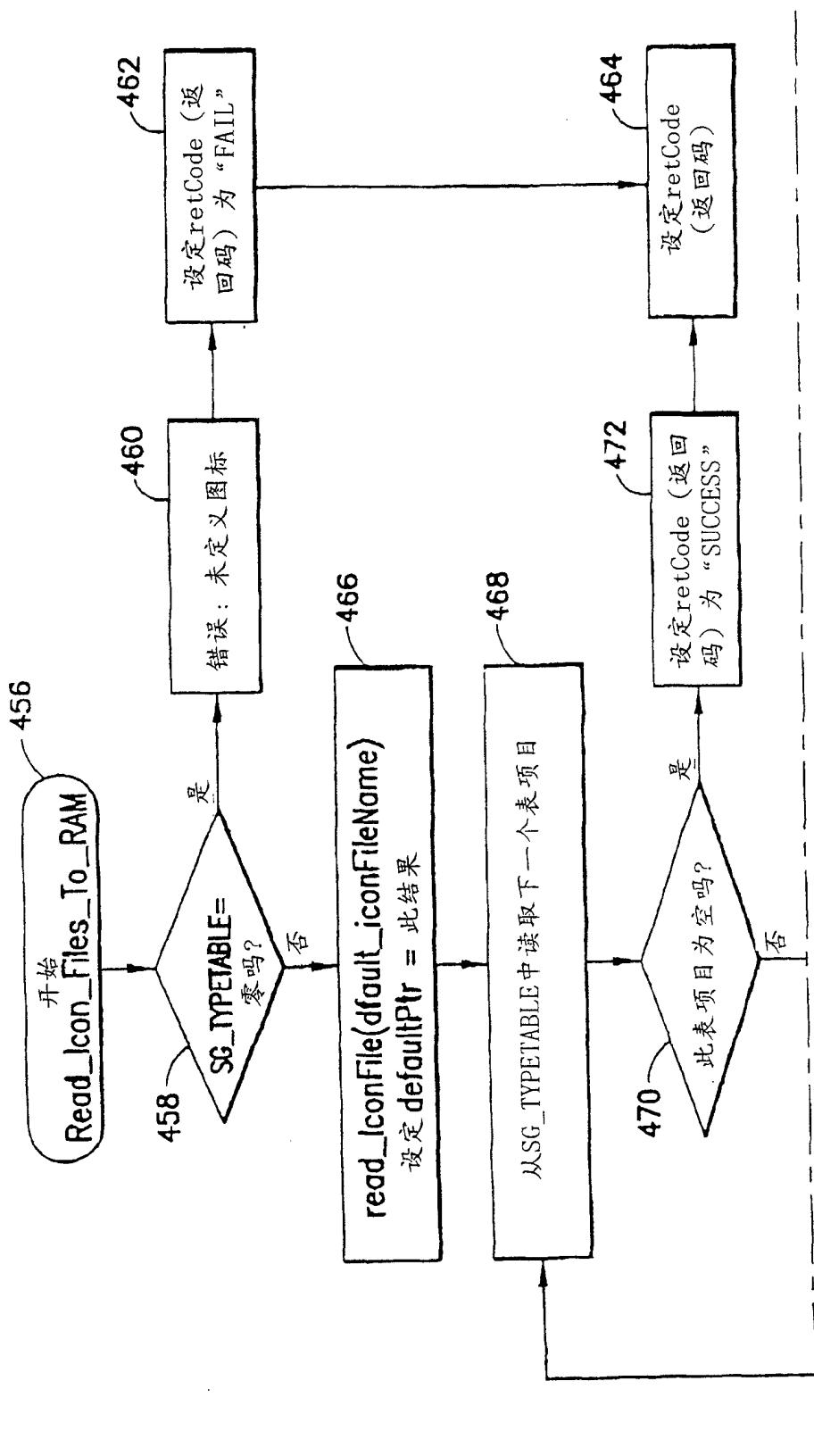
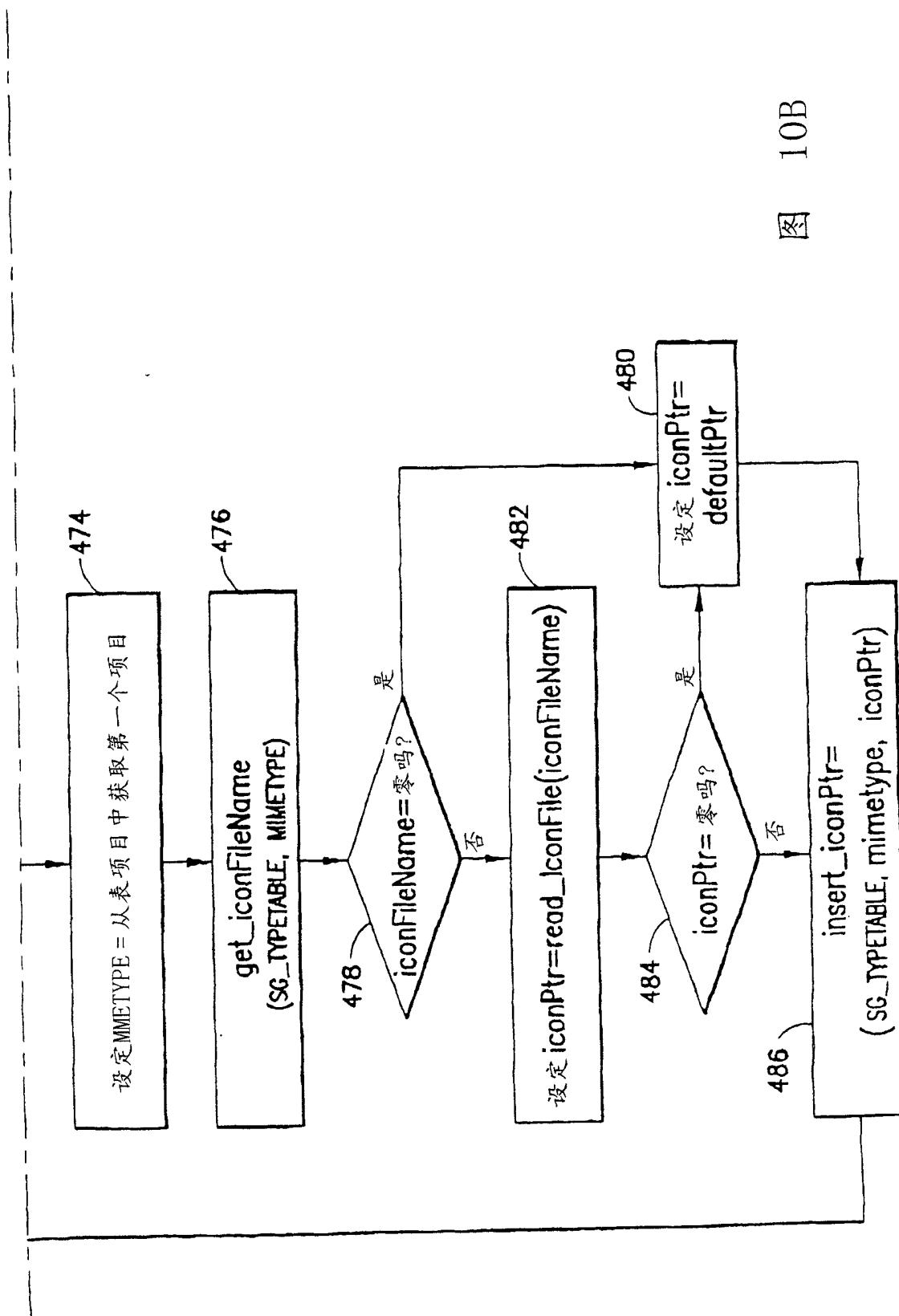


图 8









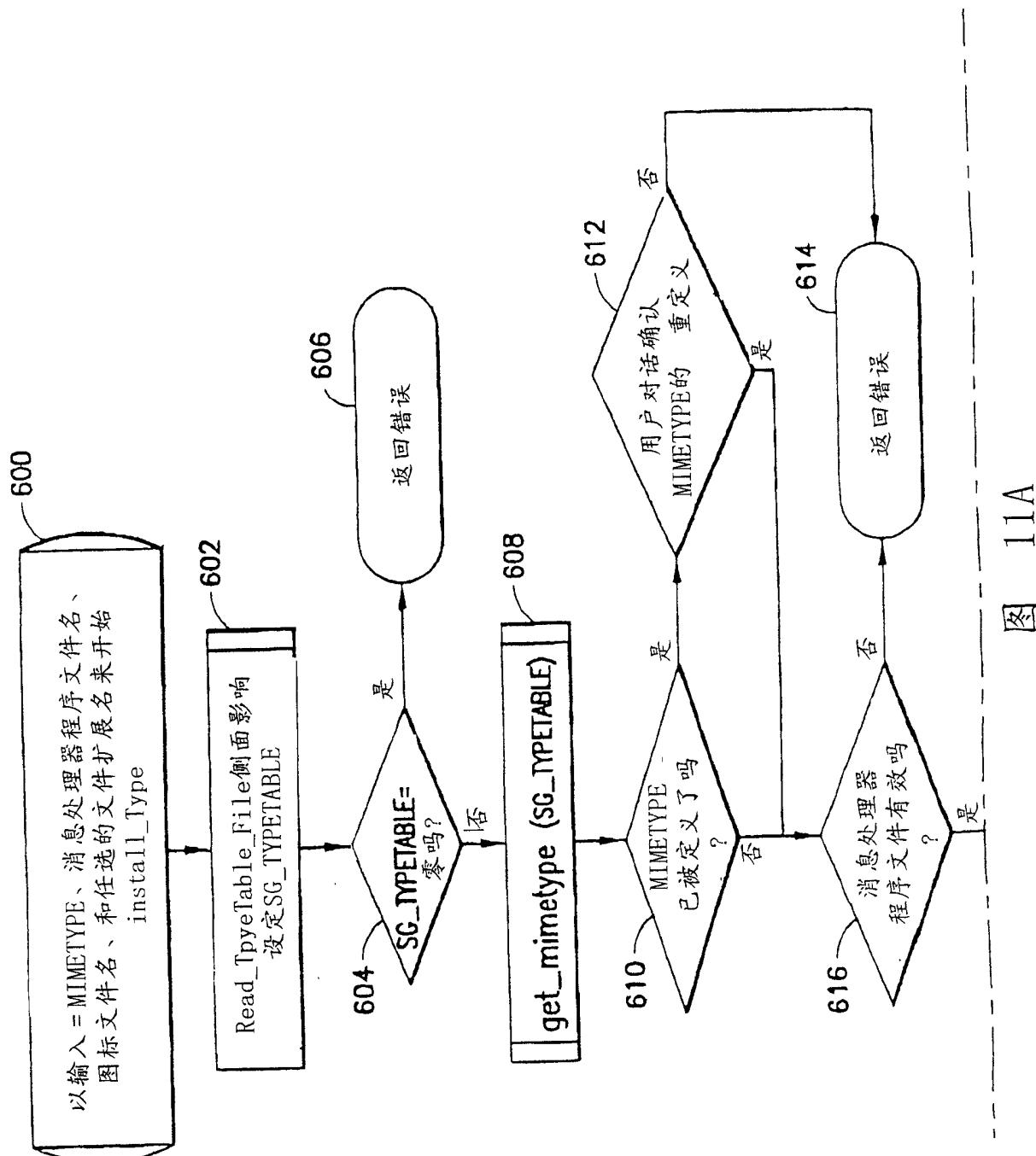


图 11A

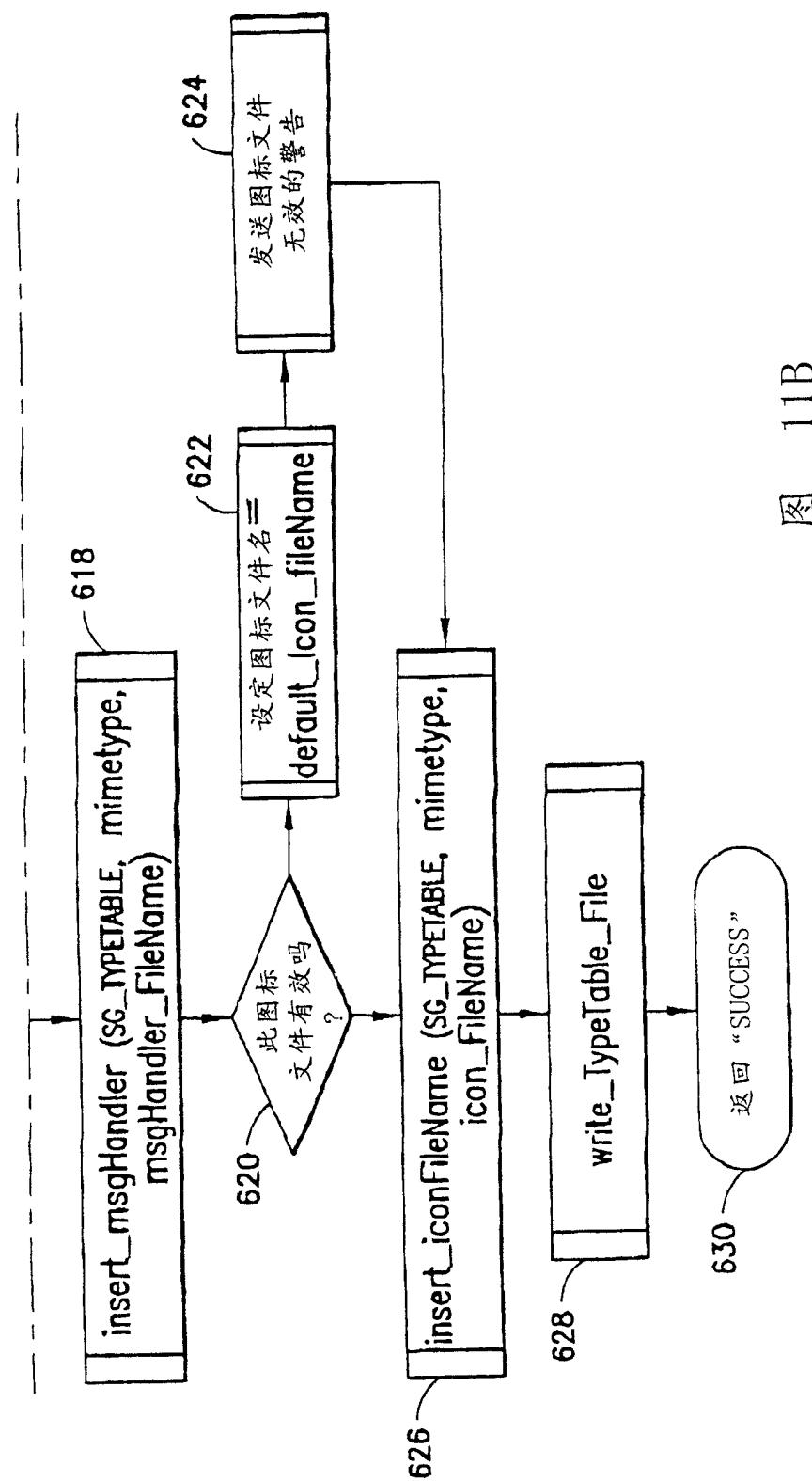


图 111B